9 ADDRESS 13 LETTER NOTE 15 STANDARDS & ERRORS RE 59 & 60 22 33 34 MAE
39 45 52 56 64 69 77
87 COMPUTER 87 DATACOMPUTER 96 COMPUTER DATA

June 1983 AN EMAP PUBLICATION 75p

Electronics & Computing MONTHLY

Britain's **First** Electronics & Computer **Applications** Magazine

SPECTRUM IMAGE PROCESSOR

store and display video signals on your Spectrum

POWER CONTROL FOR MICROS — three versatile designs

SPEECH AND MUSIC GENERATION — the techniques involved

FLOPPY TAPE SYSTEM FOR THE BBC — an in-depth review

VIC-20 SOFTWARE UPDATE

Electronics & Computing Monthly
Scriptor Court,
155 Farringdon Road,
London,
EC1R 3AD

EDITORIAL
01-833-0846

Editor
Gary Evans

Editorial Assistant
Ann Houghton

ADVERTISING
01-833-0531

Manager
Claire Fullerton

Deputy Manager
Alun Evans

PRODUCTION
01-833-0846

Design
Terry Gibbins

Make-up
Time Graphics

Publisher
David Raven

Publishing Director
Peter Peskett

Distribution
EMAP
National Publications

Published By
EMAP Business And Computer Publications

Printed By
LSG Lincoln

Subscriptions
Electronics &, Computing Monthly, (Subscription Department), Competition House, Farmdon Road, Market Harborough, Leicestershire.

ABC
MEMBER OF THE AUDIT BUREAU OF CIRCULATION

Vol. 3

IN THIS ISSUE

Issue 6

Electronics & Computing Monthly is normally published on the 13th day of each month

EDITORIAL

Some months are destined to be more eventful than others and the one just passed has certainly been a lively one for *Electronics & Computing Monthly.* Not only have I taken over from David Raven as Editor but, by the time you read this, we will have moved from our offices in Daventry to a new address in the heart of London.

All these changes will not however affect the character of the magazine that has been established in the first two years of its life. David Raven will continue to play an active part in the publishing side of *E&CM* and his unique insight into the dealings of the computer trade will help to keep our News and Opinion sections 'well informed'. Ann Houghton, who has done such a good job of organising our Software features will also continue on the Editorial side of the magazine.

Onward And Upward

We hope the above phrase will be an accurate reflection of our circulation over the next few months, and to that end, in addition to maintaining the established character of *E&CM,* we have also got some exciting new features at the planning stages. In addition there are some thoughts on the presentation of the magazine, but they are still very much under wraps at the moment.

It is going to be a very busy summer for myself and the rest of the *E&CM* crew and I hope that you will share in the excitement that we all feel in the exciting prospects of **doing** things with your computer.

Ever Downward

Not, we hope our circulation this time, but the price of computing hardware. Sinclair's latest advert in this issue shows that 'At A Stroke' the price of both the 16K and 48K Spectrum has been significantly reduced. I suspect, though that even this reduction, will leave enough money in Clive's pocket to fund the continuing development of his electric car and still leave more than enough change to keep his more conventional wheels in fuel and oil. There really is an awful amount of money to be made if you manage to get things as 'right' as Mr Sinclair has over the last few years.

Any bets on a ZX83 before the end of the year – but I guess the Microdrive should come first.

David Raven managed to get over to Germany during the month and returned with encouraging news on the price of printers. He was quoted prices of around £50 in OEM quantities. This would mean a retail price of around the £100/£150 level. Let's hope we see this sort of price soon. Many other areas of the hardware market, with mass storage systems being the most obvious, should see a continuing drop in 'Real' terms during 1983 making this the year of not only the low-cost computer but of the low-cost computer system.

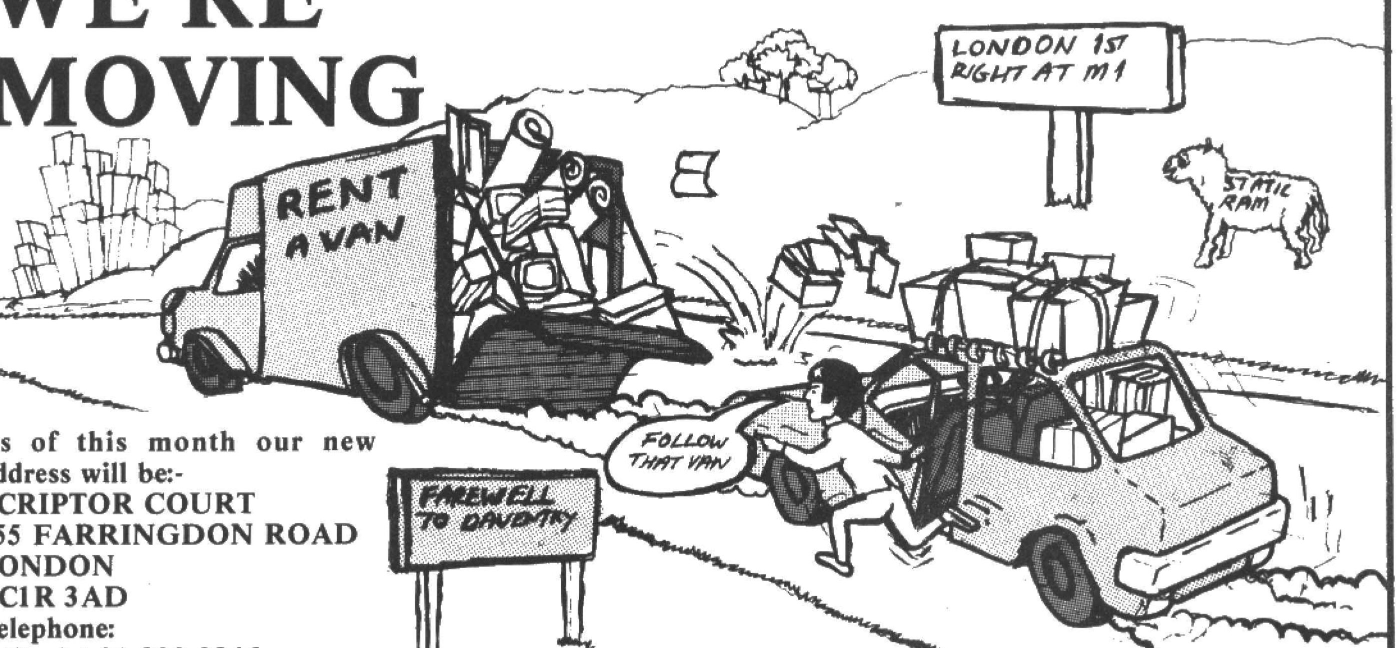
Get The Balance Right

To round off just a few words about the editorial balance of *E&CM.* We try to provide a mixed assortment of both hardware and software items but with so many different tastes to please, it is inevitable that some of you will find that some months offer less of immediate interest to you than others.

We are always pleased to hear your views on our content, and if you have **anything** that you think we are doing wrong – write and let us know.

GARY EVANS

WE'RE MOVING



As of this month our new address will be:-
SCRIPTOR COURT
155 FARRINGDON ROAD
LONDON
EC1R 3AD
Telephone:
Editorial 01-833-0846
Advertising 01-833-0531

Dear Sir,

I have been receiving E&CM for a year now (it doesn't seem that long, but after not receiving an issue for nearly two months, I realised my subscription had expired), and have in general enjoyed the magazine very much. I have a ZX81 with 16K memory expansion, and of course look forward to hardware and software projects.

I do find that there seem to be too many errors in some of the program listings and the reproductions of the ZX81 printouts are poor. Fortunately I can usually work out the problems, but I have had 10 years of programming experience.

In the hardware articles, I have frequently found typographical errors which completely baffle me, as I have not the practical experience to work many of them out. Another problem is the use of specific type numbers for some components without the required specs as often I can't find equivalents for the type number but could likely find an equivalent from the nessary specs.

Now for a specific project, the Function Key for the ZX81 in the November 1982 issue. A very useful item, but A. Fishpool must have been lucky when he built it. The Shift signal must go low before the New Line signal in order to get the Function function and if the New Line diode D11 happens to conduct a little faster than the Shift diode D10, nothing happens. My first choice of diodes didn't work, but just interchanging them solved the problem.

I have sent in my renewal, and hope you can keep up the good work with some improvement in quality control.

Yours truly,
Richard Hill,
Nova Scotia,
Canada.

We accept that errors creep in on our projects, and will generally be tidying up both presentation and quality control over the next few months.

As to software, all our published material is extensively checked and most problems when using our listings stem from "finger trouble" at the user's end. We're looking at ways of improving the quality of ZX81 and Spectrum listings – watch for an improvement soon.

Thanks for the comments about the ZX81 Function Key – we hope that anyone else who has experienced problems with this project will find your comments useful.

In Control

Dear E&CM,

I have just purchased a BBC Model B micro computer and having read most of your articles concerning the BBC, mainly "INTERFACING THE BBC MICRO", and would be very grateful if you could send me details on controlling motors as a group of friends and myself are designing a small computer controlled robot.

If it is not possible for you to send me any details, could you please send me the address of someone who could send me circuit diagrams of the interfaces needed to control the speeds of various motors.

Yours sincerely,
L. Shields,
Halifax,
West Yorks.

Well Mr. Shields – we hope the article on power control for micros will be of some help to you. In addition a project aimed specifically at the BBC machine will appear in a couple of months time.

Helping Hands

Dear Sirs,

May I say thank you for publishing my letter re problems with the article on A/D converter for Spectrum (John Taylor).

I have received help and advice from the author and a number of other people.

Perhaps a poll/questionnaire of your readers as to their age/qualifications/interests would help article writers (and yourselves) to aim at a known target of reader ability/interest.

Please keep articles on the interconnection between electronics and MPUs, (I dislike the word computer – it conveys too narrow a meaning to such a versatile machine), together with their application to monitor/report and control the outside world.

This is part of the future of our country.

Yours faithfully,
B. Hoggarth,
Kettering,
Northants.

It's nice to hear that you've received the help and advice you needed, Mr. Hoggarth. Our readers are a very helpful bunch.

As to a questionnaire, our April issue contained just such an item, although it was only included in a restricted number of issues. We're just analysing the results now and we'll be undertaking any necessary 'fine tuning' over the next few issues.

Computer Club Call

The majority of computer clubs seem to cater for **our** kind of reader, that is people who like to be at the sharp end of computing. If your club is organising any special events in future months, write and let us know. Even if there's nothing special planned, perhaps club secretaries would like to invest the odd 16p and a few minutes of their time to tell us about their clubs with details of any specific areas of interest. We'll be more than happy to publish any such information as a regular feature of E&CM.

Send all such items to the Editor at:-

Electronics & Computing Monthly,
Scriptor Court,
155 Farringdon Road,
London,
EC1R 3AD

marking the envelope 'Computer Club Call'.

Please note

We would like to point out that all letters requiring an individual reply **must** be accompanied by a SAE.

Creative Sord

One of the latest contenders in the £100-£200 price range of computers is the Sord M5. The Sord name may not be familiar to many of you but this Japanese company have had a UK presence for some time now although their product range until the M5's launch has consisted of a range of business orientated machines.

The M5 is Z80 based (4MHz version) with an 8K on board monitor in ROM. The machine also incorporates the 4 channel Z80 Counter Timer IC and makes use of two Texas ICs to take care of the video and sound functions.

The SN76489 provides 3 sound channels (making chord generation possible) plus one noise channel. Seven pre-programmed noises are also available.

The video controller is the games orientated TMS9918 device. This provides 16 display colours and four distinct screen modes. Graphic mode 1 displays 32 columns x 24 lines with an 8 x 8 dot matrix; text mode A 6 x 8 dot pattern in 40 columns x 24 lines; multicolour mode 64 columns x 48 lines as a 4 x 4 dot pattern and graphic mode 2 a full graphic display of 256 x 192 picture points.

The video controller is a sprite based device. This allows a graphic display to be built up on a number of planes (sprites) allowing quite complex animations to be built up and readily manipulated.

Another feature of the TMS9918 is that its 16K screen RAM is distinct from any user RAM. Thus, although the available user RAM on the basic machine is only 4K, as there is no conflict between this and display RAM, fair size programs can still be accommodated.

Other standard interfaces include a 2000 bit/second cassette interface, with motor on/off control, a centronics type printer output and a 56 way edge connector making the machine's bus available to the user.

This bus will in general be used to connect cartridges to the basic Ms. These will take the form of games packages, expansion RAM packs (expandable up to 32K in 4K increments) and additional ROM.

I, G or F

The M5 is unusual in that only the monitor is on board the machine being supplied with BASIC-I in a ROM cartridge. The I signifies instruction level BASIC and this, fairly standard implementation of BASIC, is fairly restricted in its scope, being designed for the first time user.

A BASIC-G cartridge (at about £35) is the Graphics language that allows the full power of the video controller to be utilised. This version of BASIC allows character blocks on the various sprite levels to be manipulated to detect collision between blocks etc.

A third cartridge carries BASIC-F, the F this time presumably signifies Floating-point, and this is billed as the maths cartridge.

The Ms also supports FALC, a simpler version of Sord's PIPS (Pan Information Processing System). This is a powerful and easy to use language that allows tyro programmers to get sophisticated data manipulation programs up and running, in a fraction of the time it would take to structure an equivalent BASIC program.

The Ms 'looks good', its construction being fairly sturdy. The keyboard is Spectrum like, although the keys have a better feel to them.

At £189.95 the Sord is at the top end of the price bracket mentioned above yet Sord are predicting a 20% UK penetration in the low cost computer market after one year. This optimistic prediction has a good chance of being realised if, as Sord hope, the Ms will be available in good outlets by the middle of the year.

Another fact that will help the Sord to quickly gain a high profile is the fact that the company have sponsored the musical Bugsy Malone to the tune of £35,000. An interesting move this, for not only will Sord achieve a fair amount of good PA from the investment but, if Bugsy is a success, they stand to make a fair deal of money as well. (If you're interested Bugsy gets under way at London's Her Majesty's theatre on May 26).

The Sord should sell well, as a games machine that is also capable of upgrading to a good general purpose machine, courtesy of memory expansion, BASIC-F and 17 centronics output. The only reservation in some people's mind will be the miserly 4K user RAM for the £190 odd asking price but, the 16K user RAM should be taken into account as this means that 4K goes a lot further than on most other computers.

New Standards

In order to bring E&CM into line with various international conventions covering the annotation of circuit diagrams, we shall be adopting the following conventions in future.

These may seem strange initially, yet the system has a number of advantages. Long strings of zeros when showing values are avoided as are decimal points.

We also replace the Ω sign with the letter R and the μ sign by the letter u. Both these changes make the job of our typesetter easier and mean that any of you can submit articles to us in 'house style' without having to handwrite μ's and Ω's throughout your manuscripts.

In showing the value of resistors and capacitors the following conventions are adopted:-

P (pico) = 10^{-12}
n (nano) = 10^{-9}
u (micro) = 10^{-6}
m (milli) = 10^{-3}
k (kilo) = 10^{3}
m (mega) = 10^{6}

Thus a resistor of 4700R will be shown as 4k7 (we use the letter R instead of the usual Ω, so that 470Ω will now appear as 470R).

A capacitor of value 0.47u (note that we also use the letter u not μ and do not show the F, Farad symbol) will be shown as 470n.

A 2200p capacitor will be shown as 2n2 while a small value capacitor would be shown as 4p7.

A similar sort of convention will often be used for showing test voltages. A voltage of 5.6V, will thus be shown as 5V6. The same rule applies to showing values of currents.

ECM/ZX Energy Success

When Holec Energy Ltd., gave one of their engineers permission to write an article in the April issue of *Electronics & Computing Monthly,* they did not bargain on such a swift and lucrative response.

Tony Chamier, designer of the *E&CM* ZX Energy Management System, reports that since this item appeared in our April '83 issue, his company has already received ten enquiries for their up market energy management systems, including an almost firm order valued at £35,000. In addition, Tony has also been asked to quote by a government department for a quantity of 3000 units of the Electronics & Computing ZX Energy Management System. Not bad for his first article in *E&CM.*

Project kits and separate PCBs are still available for the ZX Energy Control Unit. Write to: ZX Energy Control Unit, Electronics & Computing Monthly, Scriptor Court, 155 Farringdon Road, London, enclosing £35 for a complete kit, £6.50 for a printed circuit board and £3.50 for the program cassette.

Sorry, Trojan

An advert for Trojan in our May issue made mention of a mythical 'light socket' on the Dragon.

For 'light socket' please read 'joystick port'. We hope this error did not cause any inconvenience.

Brainwave To Boost Christmas Sales

A high-energy boost for the home computing industry's 1983 Christmas sales has been provided by the launch of Brainwave '83, the first Home Computing, Video and Electronic Family Show, to be held at the National Exhibition Centre in Birmingham on November 4, 5 and 6.

With one industry survey estimating that 25 per cent, or 235,000 of the 940,000 home computers in the U.K. were sold last Christmas, Brainwave '83 has positioned itself to take the lead in this year's Christmas bonanza. A separate survey projects that the number of homes possessing these computers will increase to 2 million by 1985. Manufacturers, dealers, distributors and retailers plan to use the show's November dates to create a huge giftshop where the whole family can try out and buy a complete range of electronic wizardry for Christmas gifts.

Adding to the excitement surrounding the show are plans for a massive advertising campaign involving television, radio, the consumer and trade press. Attention will be drawn not only to the entertainments inside the exhibition but also to the huge Fireworks Display and Bonfire which are scheduled as part of traditional Guy Fawkes Night activities.

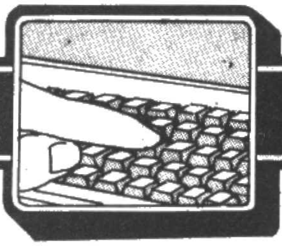
Brainwave '83 is being managed by Clapp & Poliak Europe Limited, the exhibition management company which organises the highly successful WHICH COMPUTER? Show, The Computer Trade Forum, The Scottish Computer Show and The Office Automation Show and Conference.

Mike Rusbridge, exhibition director for Brainwave '83 said that the show "will be a family wonderland, bursting with the largest range of home computers, video games and equipment, and electronic kits and accessories that has ever been assembled. The leading

INTEGRATED ERRORS

Typographical errors mean that the data shown on pages 59-60 is unreliable. A full list of corrections will appear next month.

Our apologies for any inconvenience caused.



Stir It Up?

Our second release this month is somewhat controversial. Called *"Clone"* its purpose is suggested in its name. Simply *"Clone"* will copy both headerless and standard Spectrum tapes. Its purpose is to back up tapes "purchased by the user".

We have had the program available for some time but held back from releasing it mainly due to the sudden and uncontrolled growth of software libraries operating outside of software houses' wishes.

Recently we have seen a number of these tapes advertised and decided that it would be foolish to hold back. We simply feel that if they are to be sold and it would seem they are, then we should offer the best *"Clone"* tape and let the market react as it will.

"Clone" will be sold with the following condition attached, *"Clone* is for the use of the purchaser only and is to be used solely for the purpose of making back up copies of the purchaser's software. *Clone* must not be used to produce more than one copy of the software tape and the owner may not sell, hire, or otherwise dispose of the backup copy. The purchaser is warned that purchasing *'Clone'* is an acceptance of these conditions and forms the basis of a contract to purchase. Should the purchaser knowingly infringe the copyright of any software supplier the purchaser will pay such damages as the software supplier deems reasonable".

We hope that each and every micro magazine will open discussions with its readers and existing software houses and refuse advertising for all back up tapes. However whilst advertising is accepted then we will continue to market this tape. Hopefully some discussion regarding both the users' and the suppliers' viewpoints will be aired.

We believe that *"Clone"* will back up all the current software tapes available, it will not however copy itself . . .

"Clone" is available now and costs £5.00 including post and V.A.T. *"Clone"* is available by post directly from Work Force who will hold a sales record of all purchases.

We trust we have given you something to discuss and that some good will come of it. We thank you in advance . . .

DEREK A TIDMAN.

P.S. We may call our next release "Stir It Up".

Work Force,
140 Wilsden Avenue,
Luton, Beds., England.
Tel: 0582 454456

Oric 16K OK

For those of you who ordered the 16K Oric and are at present basking in the delights of a 48K machine – the honeymoon's over.

All customers with a loan 48K model will soon be asked to swap it for the smaller 16K version. There's some silver lining to this news however as, if you want to keep the 48K computer, and who wouldn't, you'll be able to buy it at a discount of 25% making the price a miserly £127.50.

Retail stocks of the 16K machine will not be around until July but it will be available via mail order although, with such a high demand for the machines from overseas, manufacturing is at full stretch.

More Memory

VIC-20 memory expansion units for the popular VIC-20 are available in 16K or 32K chunks from Plus-80.

The company offer an impressive one year guarantee together with the further promise of a 48 hour replacement service on any faulty products.

The Plus-80 products are available by mail order and through Commodore dealers.

The company are pleased to hear from both the public and dealers, and can be found at

Unit 4,
31-33 Lower Road,
Harrow,
Middlesex, HA2 0DE.

Spectrum 32K Rampack

A 32K rampack for the Spectrum is now available from Cheetah. Initially available via mail order, the unit will shortly be retailed through a number of selected outlets.

The pack plugs into the rear edge connector and provides a painless upgrade to 48K for those of you with the 16K version. Price, including p&p is £39.95 from

Cheetah Marketing,
359 The Strand,
London, WC2R 0HS.

Glarego

Murphy's law, clause 8, sub-section 4, states that in whatever position you place your computer's monitor, the screen will be plagued by annoying and distracting reflections. One solution to this problem is to buy a tinted screen to hang over the front of your TV but these can often be difficult to affix and produce a rather unsightly arrangement.

An alternative approach is to coat the CRT with a non-reflective skin. The Glarego kit is just such a product and consists of two aerosol cans, a cloth and a strip of masking tape.

The first spray is a screen cleaner and is used with the cloth to remove any grime from the screen's surface.

The masking tape is used to protect the TV's cabinet from the second aerosol which contains the anti-glare coating.

A minimum of two coats, and up to five, are required with a period of five minutes between each coating. After 24 hours the film is dry and the screen protected from the effects of glare.

Development in Graphics Software

The 'GRAF' series of programs provided by CLARES MICRO SUPPLIES are a major development in graphics software for the BBC Micro (32K). The series includes GRAFKEY, for keyboard, GRAFSTIK, for joystick and GRAFDISK, which allows keyboard or joystick to be used for input. In addition a version for use on a graphics tablet has been developed, this version is sold with the tablet by MICREX of Alsager STAFFS. There are also plans to produce a version for a light pen if a suitable pen is found.

The philosophy behind all the programs is to provide a means of creating pictures, simple of complex, which can then be saved and used in other programs. The saving and loading is very quick for a program of this nature and using cassette the array can be saved in about 20 SECONDS or about 2 SECONDS using disk. The array provides sufficient room to draw quite complex pictures but if this is not enough you can display as many arrays at the same time as is necessary. In fact this feature provides even more flexibility because it can be used to display drawings or diagrams at various stages of completion making it ideal for teaching purposes. You can even create a standard background onto which you can add various foreground pictures, which if taken to a logical conclusion could provide the basis for animation techniques. In addition to the ordinary saves GRAFDISK provides a SCREEN SAVE which will save everything on the screen, even multiple arrays.

The FUNCTIONS provided are quite extensive and allow the drawing of Lines, Rectangles, Triangles and Circles plus Text (upper and lower case) and a colour pallette of 8 colours plus flashing. A line is drawn by plotting the start and finish point, the program does the rest. Triangles need 3 points but rectangles and circles need only 2. Circles can also be copied using just 1 point. The shapes can be either filled or just outline. A PAINT option allows the user to choose the brush size and is useful for filling awkward shaped areas but unusually it also allows just the outline of the brush to be used and this gives some very attractive effects.

DRAWING AIDS provide Alignment grid, circle copy, delete, free memory, purge memory, variable cursor speed, clear screen and REDRAW. In addition MODES can be changed whilst the program is running, choose from MODES 0, 1, 2, 4, 5.

A very special facility is RUBBER BAND which allows the user to plot a point or points and wander around the screen with lines being drawn from the cursor to the point(s). These lines are not fixed in memory until you decide that they are correct. The lines will extend and contract when the cursor is moved and allow the user to view what effect the lines will have before committing himself. It is ideal for making sure that perspective is correct and with up to 4 points allowed it can be very interesting to create abstract shapes using mirror images etc. The program is worth buying for the RUBBER BAND on its own.

A FREE "REDRAW" routine is also provided which can be used to allow the pictures created to be displayed in a host program. The data from one program can be used in others within the series.

Finally the most important feature of the 'GRAF' series is that they have been designed as open ended programs and this will allow additional processing packs to be written as needed. This will allow the users to decide what they want the program to do. Cassette based versions will be supplied as a separate tape, which can then be used to process the data. Disk users will be able to return their disk for an upgrade to the latest version and they will only be charged the difference in price between the two versions. This sort of commitment is virtually unknown at this level and is usually only provided for much more expensive progress. Plans for the first processing pack include air brush, arcs, elipses and a printer dump plus many more.

If you are fed up playing games and would like to do some serious graphics work then this may be the program. Even if you do not have a serious application it is great fun to use and its uses are only limited by your imagination. It is being used at the moment by teachers, designers, artists, a sculptor and even a meteorologist, who claims that it saves hours of work. This is one of those programs that everybody can find a use for and is recommended to all. Prices are £7.95 for GRAFKEY & GRAFSTIK or £12.95 for GRAFDISK.

SPECTRUM IMAGE PROCESSOR

Richard Sargent and Robert Harvey show that, with only a few components, it's possible to store and display video signals on the Spectrum.

The arrival of the Spectrum with its generous memory capacity and (reasonably) high resolution bit-mapped display, allows low cost experimentation in the field of Video to Computer interface. The interface board is compatible with any source of composite video such as those available from most video recorders and TV cameras. By using a technique of building the complete video picture up over a number of frames, the circuit is able to use a slow, and therefore low-cost A/D converter. Before delving into the intricacies of the video interface however, we'll briefly describe how a single frame of a TV picture is built up.

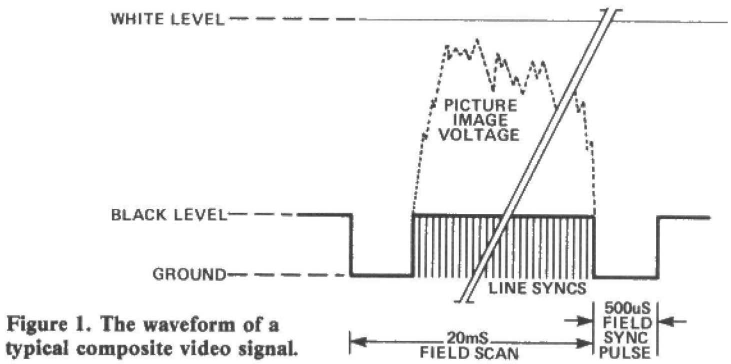


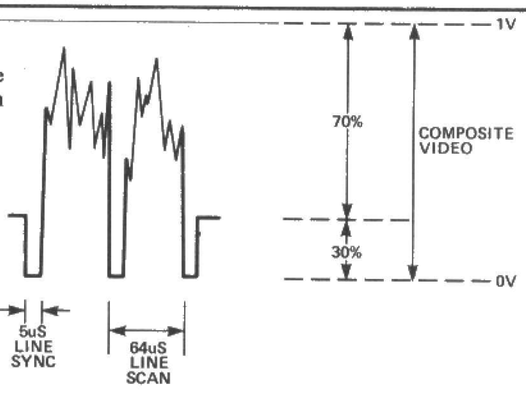
Figure 1. The waveform of a typical composite video signal.

Picture Parlance

The single point of light that creates the picture follows a regular pattern across the screen. This pattern is the television "raster" and comprises 625 lines running left to right horizontally across the face of the TV screen. The point of light, usually ever-changing in intensity, follows the pattern laid out by the lines and a complete scan along 625 lines represents one TV FRAME SCAN. In fact, each Frame scan is made of two separate FIELD SCANS, interlocking with each other. Each of these consists of 312 1/2 lines and the whole screen is scanned in 20 mS (1/50 of a second). In the next 20 mS another 312 1/2 lines are scanned, but these lines are in a slightly different position on the screen compared with the first scan. This technique is known as interlace scanning and gives a high resolution picture. Such high resolution is not our concern, so we can ignore interlacing and concentrate on how a single FIELD SCAN is built up.

Figures 1 and **2** show the waveform pattern of a typical composite video signal. If the black-level signal (no visual information present) is thought of as the base line, then it can be seen that the composite video signal is a combination of negative-going synchronisation pulses and a positive-going video voltage. A synchronisation pulse identifies the start of each 20 mS field scan. This FIELD SYNC pulse is 500 uS long and can be identified by computer software. The start of each of the raster lines is similarly marked, but with a much shorter 5 uS pulse. The time between LINE SYNC pulses is 64 uS. Of the 312 1/2 lines themselves, the first and last few do not contain picture information, and the picture information at the top and bottom screen margins is usually not important, and the software will normally concentrate on the middle 255 lines.

Figure 2. A typical TV line is a combination of a positive going video signal and negative going sync pulses.



Storing The Video

In order to store the analogue video signal in the Spectrum's RAM, it's necessary to convert the information to binary form. An analogue-to-digital converter (ADC) performs this operation, however it needs time to complete the task, and time is the one thing in short supply on a TV raster! Remember that a TV or monitor running your alphanumeric display is putting out at least 256 separate points of picture information in about 50 mS. The answer is to go after just one of those points, P1, convert it within 64 uS and then deal with a point underneath P1 on the next line scan. In this way a vertical column of points can be read into memory within the 20 ms field scan period. After the next FIELD SYNC pulse point P2, immediately to the right of P1, can be converted, as can all other points in a column under P2. The ZN427E ADC can cope with one conversion every 64 uS and will output an 8 bit number representing 256 shades of grey between white and black. If we ignore the 4 least significant bits we have a 16 shades of grey scale which should give the resolution we desire.

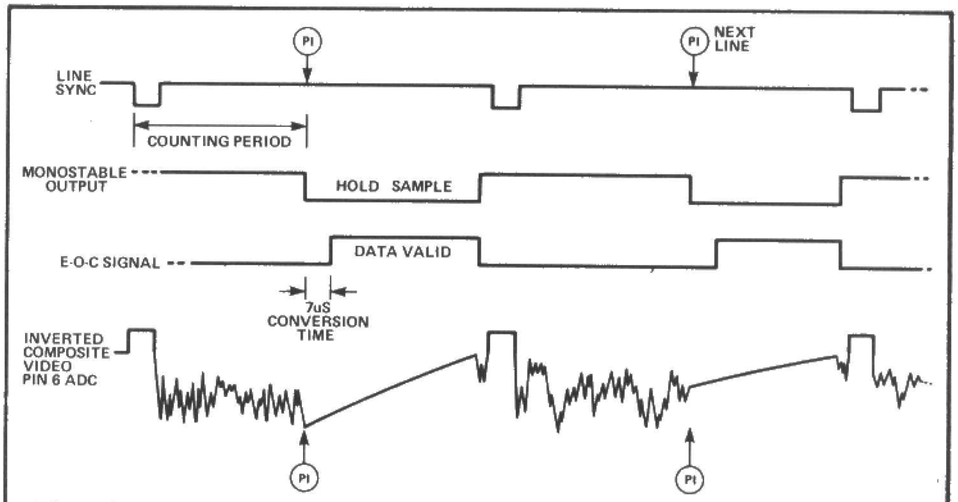


Figure 3. The processor samples one point, P1, on every line.

Experimentation Examples

To store a whole screen picture in memory will take up a lot of space, but using software control it is quite possible to select any smaller rectangular portion of the screen, and so experimentation is possible on small memory computers or on computers where a BASIC program is sharing memory with the video storage.

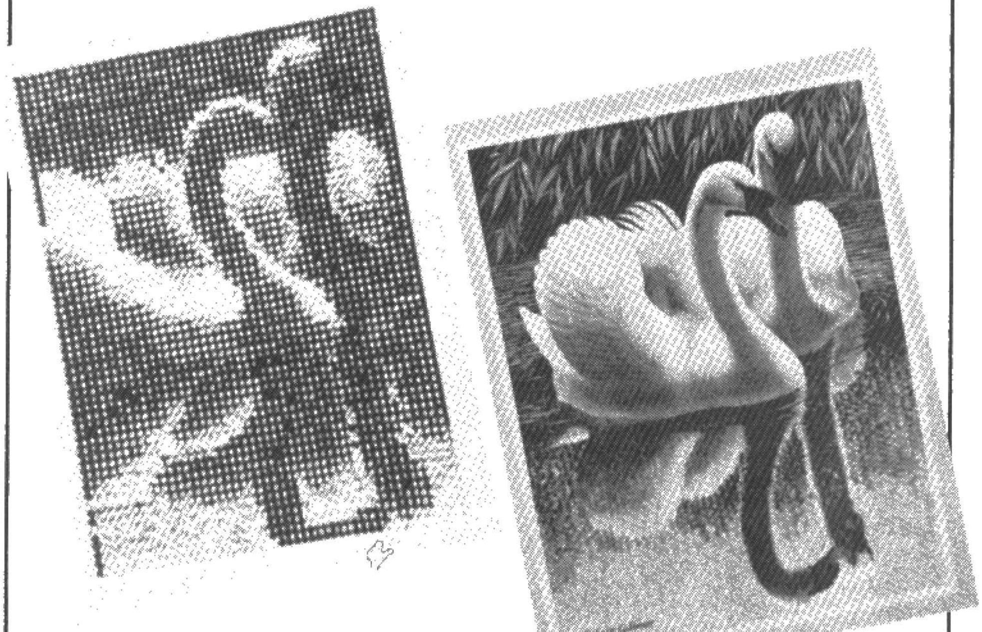
One application might be to process 32 points on 24 lines, a total of 768 readings which dould be displayed directly as character-squares on a Spectrum screen using the attributes to give 8 shades of grey.

A more demanding application would be to store 255 x 255 points and this would seem to require 65025 bytes of RAM.

However, if we are using only the 4 most significant bits from the ADC, we can pack the digital information for two picture points into a single byte and store the screen picture in about 32K bytes: easily within the capacity of a 48K Spectrum. This is an area worthy of experiment, for if very large amounts of data were being taken from the TV source, then that data would have to be packed rather more tightly into the computer memory. Eight brightness levels can be encoded in three bits, and so eight different points taken from the TV could be packed into three bytes of RAM.

The Circuit In Outline

The hardware comes in two parts. The port interface and the video interface. The port interface is of arbitrary design: its job is to write to the horizontal scan counters and to read the ADC chip and the mixed sync line. It has a secondary function of providing a centronics-



Next month in part two we'll describe a program capable of producing the high resolution output shown above.

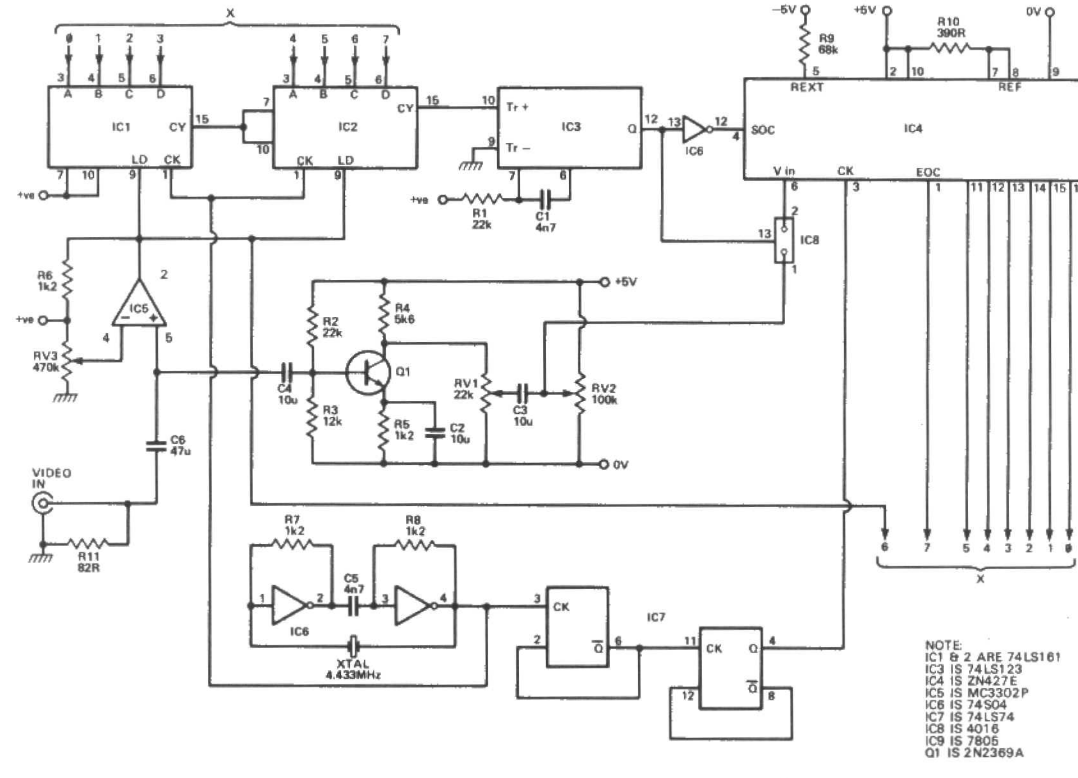


Figure 4. The complete circuit diagram of the Spectrum Image Processor. Decoupling capacitors of 100n should be added to each IC.

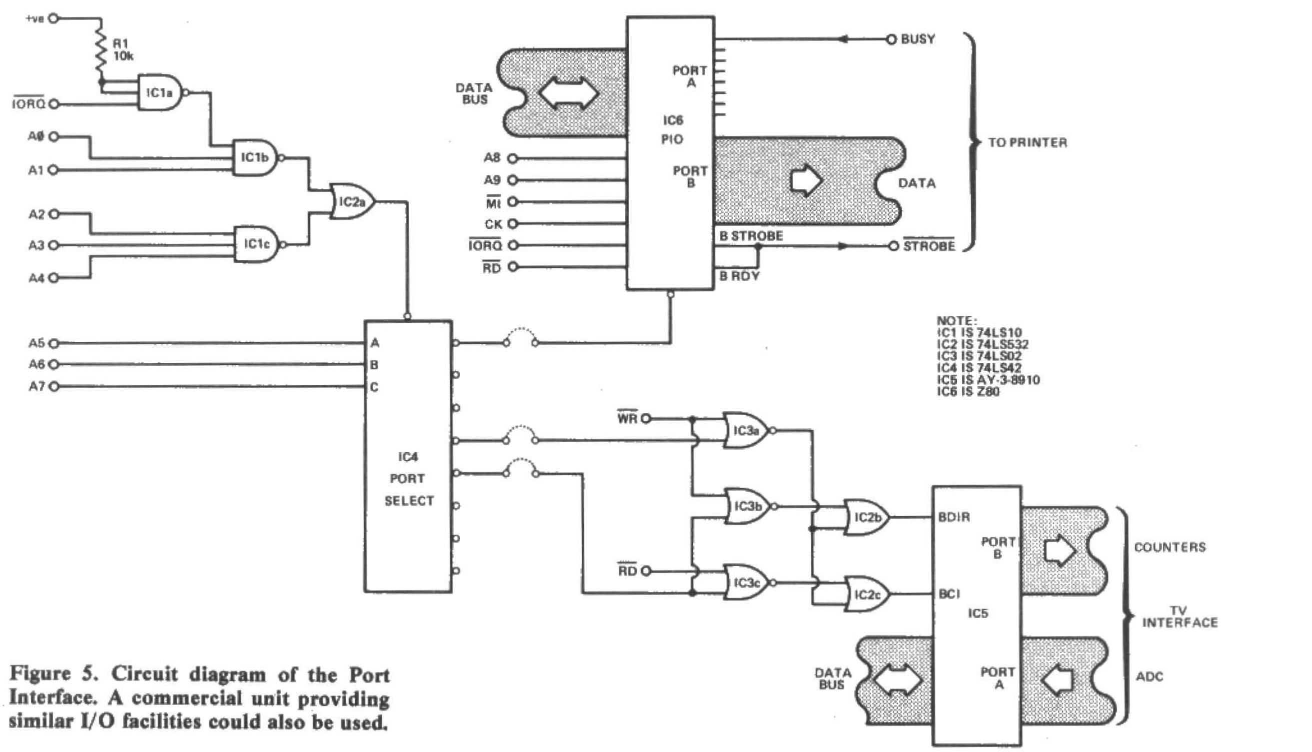


Figure 5. Circuit diagram of the Port Interface. A commercial unit providing similar I/O facilities could also be used.

standard link with a dot-addressable printer, but that's the concern of a following article. The port interface need not be built if the user possesses a commercial unit which provides similar Input/Output facilities.

The video interface has a low IC count of 8, which is a sure sign that the software is doing all the hard work. With reference to **Fig 1,** it can be seen that the composite video signal is split into its two component parts, Video and Sync (Line sync & Field sync). The Video signal is amplified, sampled and converted to a 8 bit digital word, and then tucked back into the computer. The Sync signal consists of a stream of line sync pulses which set up and trigger an 8 bit pre-settable counter, it is the pre-set value, loaded from the computer, which determines at what point along the line scan the video voltage will be sampled. When the line sync pulse has reached its HIGH level the line-scan begins, the counters start counting and when the count of 255 is reached a short CARRY pulse is generated. This pulse is stretched by a monostable to 32 uS and this causes the voltage at the sample-and-hold circuit to be held and instructs the ADC to start converting that voltage.

Meanwhile the computer is looking at the "end of conversion" output of the ADC, and as soon as this goes high the digital value of the sampled voltage is read into the program.

The Circuit In Detail

Figure 4 shows the video Interface section of the circuit and **Fig 5** shows the PIO section of the circuit. When the composite video arrives at the printed circuit board, it must be matched with the transmitting circuit. A pull-down resistor (R11) of value 75R should be used but a standard 82R value will work just as well. Without this resistor the video picture would be darkened. C6 provides DC isolation.

The composite video is passed to the comparator, IC5 (LM311), which only allows the MIXED SYNC (FIELD and LINE SYNC) to pass. RV3 is adjusted until only the SYNC pulses are detected by an oscilloscope on the output of the LM311. PORT B, BIT 6 of the port interface regularly looks at the MIXED SYNC and will detect the FIELD SYNC pulse which is 500 mS duration. This pulse is the main synchronising signal which tells the software to read into memory x samples (at one sample per line) in the 20 mS scan of the TV FRAME. The LINE SYNC pulse is taken to the counters, IC1 and IC2 (74LS161), and causes them to be loaded with the value 113, on the rising already on PORT A of the port interface, and to start counting up from that value. The counter is capable of counting from zero to its maximum value of 255 in 64 mS the duration of a single scan line. Suppose that the counter is loaded with the value 113. On the rising edge of the LINE SYNC pulse it will count from 113 to 255, and on the next count it will produce a carry signal. By this time the middle of the line scan has been reached, so the carry signal from the counter will cause the brightness level of the centre of the line to be sampled. A little under 64 mS is available for the voltage level to be digitised and stored in the computer memory. Precisely when 64 mS have elapsed, the hardware will be sampling the same point on the next line down, although the software may have instructions not to store that particular point in memory. The master clock which runs the counter is formed by two gates from IC6, the 74S04, and is a standard configuration and uses a 4.433 MHz crystal. The 74S04 is the high speed version of the 7404 and must be used if the circuit is to work properly. The clock signal is divided by four in IC7 to provide a 1.1MHz clock for the ADC. This clock frequency allows the fastest possible conversion rate for the ZN427E chip. The carry signal from the counter is merely a spike-pulse, and if it is to be of any use it needs to be lengthened by IC3, a 74LS123 monostable. The output from this IC is low for 32 mS and this opens the CMOS switch IC8 (4016) and instructs the ADC to start conversion. When the CMOS switch is opened, no new video voltages can reach the ADC and natural parasitic capacitance between the CMOS switch and the N-MOS ADC holds the voltage level constant for the 7 mS needed for the ADC chip to sample the voltage. The data outputs of the ADC become valid and the IC generates an end-of-conversion pulse. The software then has about 26 mS to detect the end-of-conversion and read the data outputs before the ADC reverts to reading the incoming video voltage. It is important that the end-of-conversion signal is switched off after 32 mS because the software will soon be looking at this pin again and must 'latch on' to it and wait for it to go high.

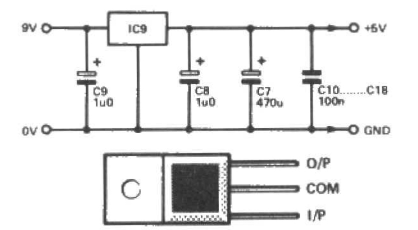


Figure 6. The image processor's power supply and the pin-out for the three terminal regulator.

The video voltage is too low to be used effectively by the ADC, being about 0.8V peak value when it enters the circuit. A single transistor amplifier having a gain of about five boosts the signal to a respectable level, but inverts the video in the process. (Software later corrects the inversion). The gain of this amplifier is adjusted using RV1, and this provides a contrast setting. RV2 provides a brightness level setting. The voltage at the amplifier side of the CMOS switch swings between zero and a positive value. A high voltage represents peak white video (since inversion has taken place). The highest voltage will give the greatest digital value on the data out pins of the ADC.

The board should be liberally sprinkled with power supply decoupling capacitors of 10n value, one for each IC. Do not be tempted to save money by omitting the on-board 5 volt power supply. The port interface can be powered from the host computer, but the video interface, small though it is, does put a strain on a Sinclair Spectrum, partly because of IC6, the 74S04, which is greedy for power.

16K Spectrum Software

The short program outlined in **Fig 7** displays what might be termed very low resolution graphics! It takes a rectangle 64 points wide and 48 lines deep and displays the resulting picture in 8 shades of grey on the screen.

```
0300
0300          ;SPECTRUM TV INTERFACE V1 (ATTRIBUTES)
0300          ;(c) R Harvey & R Sargent 1983
0300
5C08          $KEY       EQU 23560        ;Last key press store
5800          ATTR       EQU 5800H        ;Screen attribute area
FF7F          PSG$A      EQU 0FF7FH       ;PSG address-line port
FF9F          PSG$D      EQU 0FF9FH       ;PSG data-line port
000E          PORTA      EQU 14           ;Output port (to counters)
000F          PORTB      EQU 15           ;Input port (from ADC chip)
0040          SYNC$BIT   EQU 40H          ;Bit 6 provides video sync
0007          DATA$BIT   EQU 7            ;Bit 7 confirms data is valid
0050          WASTE      EQU 80           ;n TV lines which are ignored
0018          LINES      EQU 24           ;n lines to be scanned
0018                     ;EQU 48 for the MOSAIC version
0040          NCNT       EQU 64           ;Used in display routine
0002          H$DISP     EQU 2            ;Displacement between points
0020          CTR$FN     EQU 32           ;Finish of software scan
003F          CTR$ST     EQU 32*H$DISP-1  ;Scan count
003F                     ;EQU NCNT*H$DISP+CTR$FN-1
003F                     ;for the MOSAIC version
0001          DEL        EQU 1            ;30 us delay
0001
7000                     ORG 28672
7000                     LOAD 28672
7000
7000 210000   SCAN       LD HL,0
7003 AF                  XOR A
7004 32085C              LD ($KEY),A      ;Clear key-press store
7007 017FFF              LD BC,PSG$A         ;Set PSG so that
700A 3E07                LD A,7              ;PORT A is output to
700C ED79                OUT (C),A           ;the counters and
700E 019FFF              LD BC,PSG$D         ;PORT B is input from
7011 3E7F                LD A,7FH            ;the ADC
7013 ED79                OUT (C),A           ;
7013
7015 21F570   S$LOOP     LD HL,STORE      ;Store is where results
7018 1E3F                LD E,CTR$ST      ;of ADC will go : E is a counter
7018
701A F3       H$LOOP     DI               ;Disable interrupts
701B 017FFF              LD BC,PSG$A         ;Load up the 8-bit counters
701E 3E0E                LD A,PORTA          ;
7020 ED79                OUT (C),A           ;
7022 019FFF              LD BC,PSG$D         ;
7025 3EFF                LD A,0FFH           ;
7027 ED79                OUT (C),A           ;
7029 CDA970              CALL WAIT$VS     ;Wait for Frame Sync to arrive
702C 1650                LD D,WASTE       ;D counts the wasted top lines
702C
702E ED78     W$LOOP     IN A,(C)         ;Now wait for
7030 CB7F                BIT DATA$BIT,A   ;line sync
7032 28FA                JR Z W$LOOP      ;to trigger
7032                                      ;
7034 ED78     W0$LOOP    IN A,(C)         ;the software
7036 CB7F                BIT DATA$BIT,A   ;
7038 20FA                JR NZ W0$LOOP    ;and let
703A 15                  DEC D            ;it happen
703B 20F1                JR NZ W$LOOP     ;D times
703D 017FFF              LD BC,PSG$A         ;Set up 8-bit
7040 3E0E                LD A,PORTA          ;counter again
7042 ED79                OUT (C),A           ;
7044 019FFF              LD BC,PSG$D         ;
7047 ED59                OUT (C),E           ;
7049 017FFF              LD BC,PSG$A         ;
704C 3E0F                LD A,PORTB          ;
704E ED79                OUT (C),A           ;
7050 019FFF              LD BC,PSG$D         ;
7053 1618                LD D,LINES       ;D TV lines to be processed
7053
7055 ED78     V$LOOP     IN A,(C)            ;Wait for ADC
7057 CB7F                BIT DATA$BIT,A      ;data to become
7059 28FA                JR Z V$LOOP         ;valid
705B 77                  LD (HL),A           ;and then
705C 23                  INC HL              ;store it
705C
705D ED78     V0$LOOP    IN A,(C)         ;Now use
705F CB7F                BIT DATA$BIT,A   ;the rising
7061 20FA                JR NZ V0$LOOP    ;and falling
7061                                      ;edges
7063 ED78     S0$LOOP    IN A,(C)         ;of the
7065 CB7F                BIT DATA$BIT,A   ;data-valid
7067 28FA                JR Z S0$LOOP     ;bit
7067                                      ;to keep
7069 ED78     S1$LOOP    IN A,(C)         ;the whole
706B CB7F                BIT DATA$BIT,A   ;system in
706D 20FA                JR NZ S1$LOOP    ;synchronisation
706D
706F 15                  DEC D            ;Continue until
7070 C25570              JP NZ V$LOOP     ;all lines done
7070
7073 017FFF              LD BC,PSG$A         ;Set up 8 bit
7076 3E0E                LD A,PORTA          ;counters again
7078 ED79                OUT (C),A           ;
707A 019FFF              LD BC,PSG$D         ;
707D 3EFF                LD A,0FFH           ;
707F ED79                OUT (C),A           ;
7081 FB                  EI               ;Enable interrupts
7082 7B                  LD A,E           ;Check to see if all points along
```

Figure 7a. The first part of the 'low resolution' graphics routine.

```
7083 D602                SUB H$DISP       ;the line scan have been converted
7085 5F                  LD E,A           ;and if not
7086 0000                DB 0,0           ;then loop round and do some more
7086                     ;see below *
7088 D21A70              JP NC H$LOOP     ;else complete picture captured
708B 3A085C              LD A,($KEY)      ;so check keyboard
708E E65F                AND 5FH          ;and hold the routine & display
7090 FE53                CP "S"           ;if S for stop pressed
7092 2007                JR NZ NOSTOP     ;
7094 3A085C   STOP       LD A,($KEY)      ;Stay in this keyboard
7097 E65F                AND 5FH          ;loop until another key is
7099 20F9                JR NZ STOP       ;pressed, then fall through
7099                                      ;
709B CDD170   NOSTOP     CALL DISPLAY     ;
709E 3A085C              LD A,($KEY)      ;and loop around
70A1 E65F                AND 5FH          ;to build up another picture
70A3 FE42                CP "B"           ;unless B for break pressed
70A5 C21570              JP NZ S$LOOP     ;in which case
70A8 C9                  RET              ;return to ZX-MONITOR
70A8
70A8                     ;* Replace DB 0,0 with CP CTR$FN for the
70A8                     ;  MOSAIC version : The code is FE 20
70A8
70A8
70A8
70A8          ;Wait for Vertical (Field) Sync Pulse
70A8
70A9 017FFF   WAIT$VS    LD BC,PSG$A         ;Read data port
70AC 3E0F                LD A,PORTB
70AE ED79                OUT (C),A
70B0 019FFF              LD BC,PSG$D
70B0
70B3 ED78     VS$LOOP    IN A,(C)
70B5 E640                AND SYNC$BIT
70B7 20FA                JR NZ VS$LOOP    ;Loop until FS pulse present
70B9 3E01                LD A,DEL         ;Waste some time
70BB CDCB70              CALL DELAY       ;then look again
70BE ED78                IN A,(C)         ;If not still there
70C0 E640                AND SYNC$BIT     ;it wasn't the right
70C2 20EF                JR NZ VS$LOOP    ;pulse so look again
70C2
70C4 ED78     VS0$LOOP   IN A,(C)         ;Wait for
70C6 E640                AND SYNC$BIT     ;it's rising
70C8 28FA                JR Z VS0$LOOP    ;edge then
70CA C9                  RET              ;return
70CA
70CA          ;Delay (approx 10us per iteration)
70CA
70CB F5       DELAY      PUSH AF
70CC F1                  POP AF
70CD 3D                  DEC A
70CE 20FB                JR NZ DELAY
70D0 C9                  RET
70D0
70D1          FIN1       EQU $
00D1          BYT1       EQU FIN1-SCAN
00D1
70D1                     ORG 70D1H
70D1                     LOAD 70D1H
70D1
70D1          ;DISPLAY ROUTINE : Attribute version
70D1
70D1 DD210058 DISPLAY    LD IX,ATTR
70D5 21F570              LD HL,STORE
70D8 112000              LD DE,32         ;Offset value
70DB 0E20                LD C,32          ;Column counter
70DB
70DD 0618     D$LOOP     LD B,24          ;Line counter
70DF DDE5                PUSH IX
70DF
70E1 7E       D$L1       LD A,(HL)        ;Obtain brightness
70E2 2F                  CPL              ;Software invert
70E3 E638                AND 38H          ;00111000
70E3                                      ;6-bit brightness now 3-bit
70E5 DD7700              LD (IX+0),A      ;and is loaded as an attribute
70E8 DD19                ADD IX,DE        ;Step down
70EA 23                  INC HL           ;Get another brightness value
70EB 10F4                DJNZ D$L1        ;and deal with each Row
70EB
70ED DDE1                POP IX
70EF DD23                INC IX           ;Deal with next Column
70F1 0D                  DEC C            ;and continue setting
70F2 20E9                JR NZ D$LOOP     ;attributes of each Row
70F4 C9                  RET              ;until finished
70F4
70F5          FIN2       EQU $
0024          BYT2       EQU FIN2-DISPLAY
0024
70F5          STORE      DS 24*32
70F5
73F5          FIN4       EQU $
0300          BYT4       EQU FIN4-FIN2
0300
0300
0300
0300
0300
```

Figure 7b. The graphics program continued.

The horizontal loop is time critical and therefore the interrupts must be disabled. The main side effect of this is to disable the keyboard, which is interrupt driven. This is not a disadvantage in this case, providing that the interrupts are enabled at a later stage.

The vertical loop contains a lot of code to operate the PSG chip which is used as the input/output device. The use of the Z80 PIO would make things simpler, as would using shorter input/output instructions such as 'out 254,20', rather than the longer version which outputs via the BC register. The program works however, and that's the object of the exercise. It does mean though, that there is not really sufficient time to process the data from the ADC, so this is stored in main memory immediately. In the program presented, packing the data is not necessary because only a very modest amount of information is being collected. Once collected, the DISPLAY routine takes over and sets about processing the information and pushing it to the screen.

Loading And Running The Program

The object code from the source listing can be entered directly into the Spectrum if you have a machine-code loading routine. If not, then the BASIC program **(Fig 8)** can be entered and run, and this will load the machine-code routine for you. The program is run with RANDOMISE USR 28672. Pictures will be displayed upon the screen every few seconds if all is well. Key press S will freeze a particular picture, and B will break out of the routine. When using a TV camera, a high level of lighting is required and subjects with good contrast give the best results. The setting of potentiometers RV1 and RV2 can alter the picture considerably, and patience is needed to get the levels right for a particular subject.

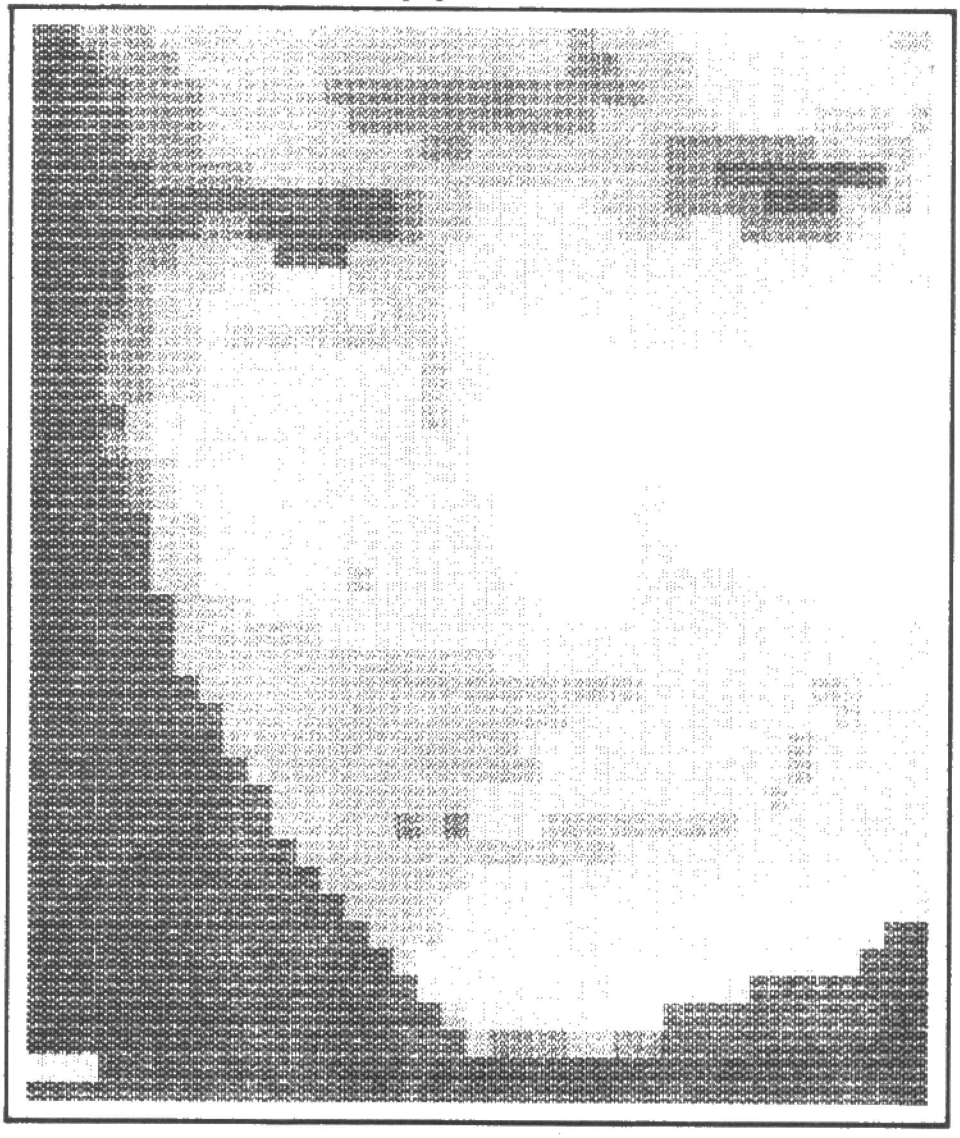
Next month we will describe a system capable of higher resolution than that afforded by the Spectrum ATTRIBUTES (colourful though they are). The method used will allow the Spectrum screen picture to be sent to a dot matrix printer and will allow the screen picture to be further processed by BASIC software.

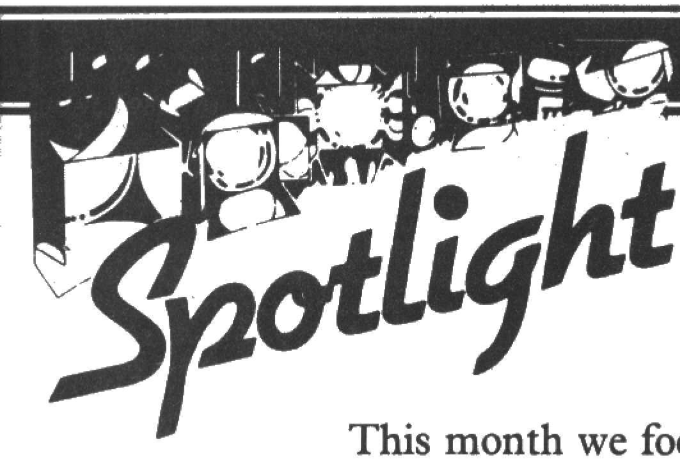
E&CM

```
1000 CLEAR 28415
1010 LET A = 28671
1020 LET A = A + 1
1030 READ D : POKE A,D
1040 IF A < 28917 THEN GOTO 1020
1042 PRINT "Start cassette"
1044 PAUSE 0
1050 SAVE "core" CODE 28672,245
1060 STOP
1070 REM ATTRIBUTES ROUTINE
1072 DATA
 33 , 0 , 0 , 175 , 50 , 8 , 92 , 1 , 127 , 255 , 62 , 7 ,
 237 , 121 , 1 , 159 , 255 , 62 , 127 , 237 , 121 , 33 , 245 , 112 ,
 30 , 63 , 243 , 1 , 127 , 255 , 62 , 14 , 237 , 121 , 1 , 159 ,
 255 , 62 , 255 , 237 , 121 , 205 , 169 , 112 , 22 , 80 , 237 , 120 ,
 203 , 127 , 40 , 250 , 237 , 120 . 203 , 127 , 32 , 250 , 21 , 32 ,
 241 , 1 , 127 , 255 , 62 , 14 , 237 , 121 , 1 , 159 , 255 , 237 ,
 89 , 1 , 127 , 255 , 62 , 15 , 237 , 121 , 1 , 159 , 255 , 22 ,
 24 , 237 , 120 , 203 , 127 , 40 , 250 , 119 , 35 , 237 , 120 , 203 ,
 127 , 32 , 250 , 237 , 120 , 203 , 127 , 40 , 250 , 237 , 120 , 203 ,
 127 , 32 , 250 , 21 , 194 , 85 , 112 , 1 , 127 , 255 , 62 , 14 ,
 237 , 121 , 1 , 159 , 255 , 62 , 255 , 237 , 121 , 251 , 123 , 214 ,
 2 , 95 , 0 , 0 , 210 , 26 , 112 , 58 , 8 , 92 , 230 , 95 ,
 254 , 83 , 32 , 7 , 58 , 8 , 92 , 230 , 95 , 32 , 249 , 205 ,
 209 , 112 , 58 , 8 , 92 , 230 , 95 , 254 , 66 , 194 , 21 , 112 ,
 201 , 1 , 127 , 255 , 62 , 15 , 237 , 121 , 1 , 159 , 255 , 237 ,
 120 , 230 , 64 , 32 , 250 , 62 , 1 , 205 , 203 , 112 , 237 , 120 ,
 230 , 64 , 32 , 239 , 237 , 120 , 230 , 64 , 40 , 250 , 201 , 245 ,
 241 , 61 , 32 , 251 , 201 , 221 , 33 , 0 , 88 , 33 , 245 , 112 ,
 17 , 32 , 0 , 14 , 32 , 6 , 24 , 221 , 229 , 126 , 47 , 230 ,
 56 , 221 , 119 , 0 , 221 , 25 , 35 , 16 , 244 , 221 , 225 , 221 ,
 35 , 13 . 32 , 233 , 201 , 26
```

Figure 8. (Above) the BASIC machine-code loader program.

Typical output of the low resolution program.





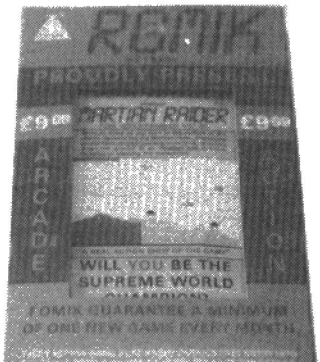
Spotlight

This month we focus on the rise and rise of Romik Software.

Romik Software have been trading for a little over six months, yet have in that time become one of the leading software houses in the UK.

Mike Barton, one of Romik's co-directors explained to us that he first thought of forming the company back at the start of 1982. At that time he had just bought a VIC-20 and had begun a job as a sales manager for a software house. He explained that his interest in games software soon grew as did his frustration with the products on the market at that time and his disillusion with his employers.

The obvious action was to set up his own company.



With the help of Gerry Rose, an old friend and Steve Clark, a successful software author about to embark on a computer studies degree at the University of London, he planned to launch Romik at the 82 PCW Show.

In the three weeks prior to the show, the three collected together a number of marketable programs and arranged for the all important packaging and literature to be produced. Rather than test the market via mail order, Romik made the decision to form a dealership network from the word go. This underlines the confidence the directors have in their products.

The company's launch was successful and since PCW the number of outlets for Romik software has increased to over 1000 and is still growing at a rate of about 100 a month!

The initial software was designed for the unexpanded VIC-20 and Romik still lead the market in self produced games material for this machine thanks to the continued demand for such programs as 'Martian Raiders' and 'Multisound Synthesiser'.

The company has however recently launched programs for the ZX81, BBC and Dragon micros and has software for the Spectrum (16K), Commodore 64, Atari 400 and 800 and the Lynx in the pipeline. The company's aim is to release at least one game every month.

Mike Barton doesn't believe in market surveys but thinks that if a game is both original and of high quality, it will sell.

Romik have 16 machine-code programmers on exclusive contract, who not only conceive their own programs but also vet any program submissions. Most games received by Romik require fairly extensive rewriting before reaching the company's standard.

Once the decision to market a game has been made, a minimum of 2000 tapes are produced together with the essential high quality packaging. Romik strongly believe that the tapes' packaging should include 'Action Shots' of the game.

Strict quality control is maintained during production of tapes and this accouts for the very low number of tapes returned to Romik. In fact, to date, only 50 tapes have been returned and of those only nine were found to be faulty, the remaining tapes were returned to the customer as being OK.

All Romik tapes are sold at £9.99 – a price people are prepared to pay for the quality of product that Romik produce.

The company has expanded into overseas distribution with their tapes being marketed in 25 countries. They are also manufactured under licence in 5 countries including the USA and Australia.

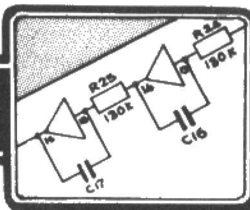
Romik expect that at least three of their programs (Martian Raiders, Moons of Jupiter and Multisound Synthesiser) will have sales that individually top 20,000 during 1983.

In only six months Romiks growth has been staggering yet, claims Mike, always under control. Even he though, might not have predicted that the company would soon be moving to new premises ten times the size of their present home.

STOP PRESS

Romik's new address is now:
272 ARGYLE AVENUE, SLOUGH.

E&CM



Physics Experiments

Dr. Alastair Allen of Norwich City College with some practical physics experiments with the BBC computer.

Voltage-current characteristics occur as part of most introductory physics and electronics courses. They are often the subject of student experiments, for example plotting diode characteristics. These graphs are less often the subject of demonstrations. However, with a micro and two analog to digital converters, these characteristic curves can be drawn rapidly, forming part of an effective demonstration.

Using two channels of the BBC Micro's converter, it is simple enough to measure voltage and current. The basic idea is shown in **Fig 1.** This is the simplest configuration, in which voltages up to the reference voltage of the converter are measured directly. A variable voltage is applied to a component under test, the PD across which is measured with channel 1 of the ADC. In order to measure the current, the voltage across a series resistor (R_S) is determined. It is important to remember that you can have only one ground point, and only positive voltages are converted. So the voltage across R_S is given by ADVAL(2) – ADVAL(1). Because we are subtracting two measurements, the experimental uncertainty in the current is larger; but this circuit is easy to set up and gives perfectly acceptable results.

Program 1 allows you to draw V-I graphs using this arrangement. The user is asked for two pieces of information. VMAX determines the (horizontal) voltage scale: this enables you to choose a sensible range of interest. The value of the series resistor is entered so that the current may be calculated. VIPLOT displays the values of voltage and current and plots the corresponding point. The values are calculated on the basis that the reference voltage of the ADC is 1·82 V. This is the value at which it seems to stabilise when the computer has been on for a few hours: it is more like 1·86 V when starting from cold. Pressing the space bar enables the ranges of voltage and current to be changed.

Figure 2 shows some typical plots obtained: a resistor and two diodes, showing nicely the higher turn-on voltage of silicon over germanium.

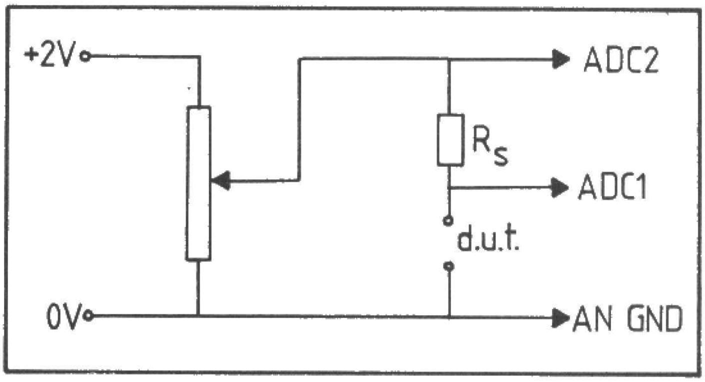


Figure 1. Simple circuit for measuring voltage and current.

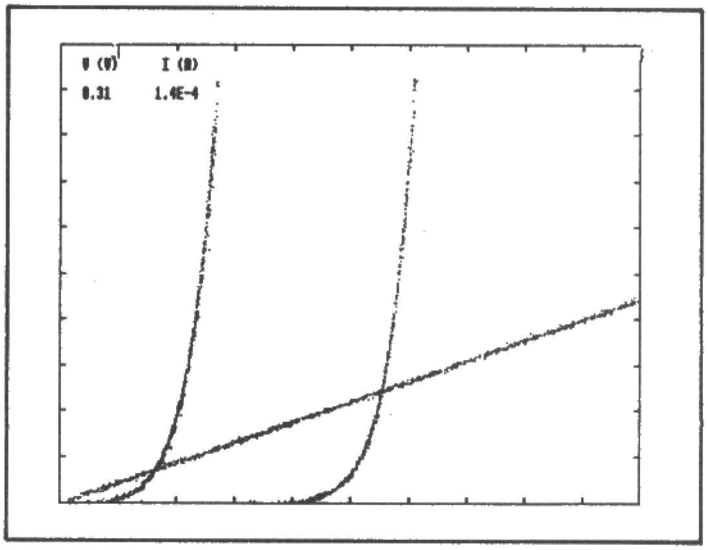


Figure 2. Graphs of current vs. voltage plotted using the BBC Micro.

Op amps

If a larger range of voltages is to be measured, then probably the best way is to use operational amplifiers on the converter inputs. This also enables you to measure negative voltages.

Obviously it is important that the measuring circuit disturbs as little as possible the circuit being measured. The input impedance of each channel of the converter is about 10 M, so this will be alright unless circuit resistances are very high. Once we start using op amps, we must ensure that the input impedance is sufficiently high. This is achieved with a unity gain buffer, either non-inverting or inverting **(Fig 3).** The input impedances in these two cases are the op amp input resistance (2 M0 for a 741) and R_B, respectively.

The buffer is followed by a simple inverting amplifier. This has the advantage that the gain may be less than one, so large

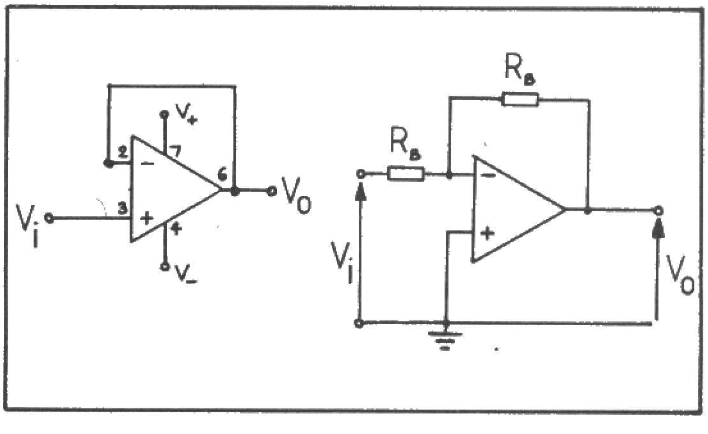


Figure 3. (a) Non-inverting, (b) inverting unity gain buffers.

voltages can be reduced to a range that the converter can handle. **Fig 4** shows some possible values of resistors which give useful ranges of operation. The complete arrangement is shown in **Fig 5.** The voltage across the component is given by ADVAL(1), and ADVAL(2) reads the current.

Using this set-up, voltages up to 15 V (the limit set by the op amp) can be scaled down. **Fig 6** gives one possible application: the characteristics of a light bulb, showing the typical hysteresis on cooling.

Other applications include such things as plotting successive curves for a diode at a series of temperatures, or a photodiode under various illuminations. As it is formulated here, this is essentially a demonstration, and as such it needs to be simple to set up and to have an uncluttered display. However, it could be adapted to use as an experiment, with individual data points being taken, and a best line fit made to the points.

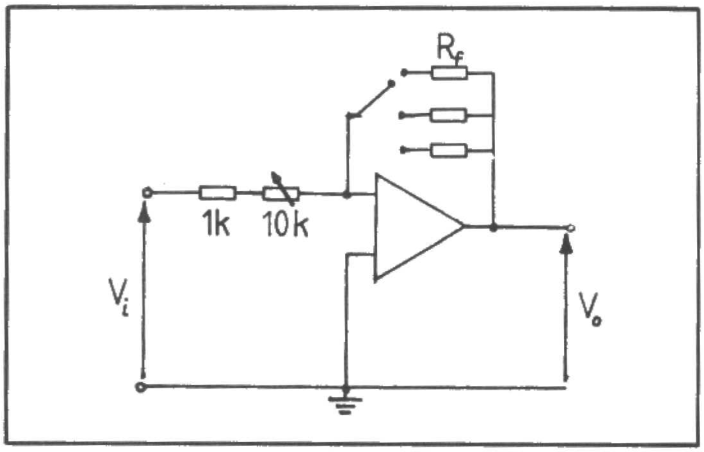


Figure 4. Inverting amplifier. Gain A = Rf/R1. Rf = 1k, 10k, 100k.

Storage Scope Applications

There are several interface units coming onto the market that enable the BBC Micro to be converted into a digital storage oscilloscope. A design to build your own interface was published in the December '82 *Electronics and Computing Monthly.* The ability to store the data from a transient phenomenon or simply to use it as a display oscilloscope, is of value in many areas of science teaching. A few of the possible applications in physics are indicated here.

Transient oscillations in an LCR circuit may be set up when a voltage step is applied, as shown in **Fig 7.** The output voltage is given by

$$V_O = V_i \left[1 - \exp\left(\frac{-R}{2L}t\right) \cos wt\right]$$

where $w = \sqrt{\frac{1}{LC} - \left(\frac{R}{2L}\right)^2}$. If the system is to be under damped, i.e. for it to ring when a step input is applied, the w»1. **Fig 8** shows such a transient, using the 0 to +10 V input to the converter. The values used in this case were L = 1 H, C = 100n, R = 100R, and V_i = 5 V.

Another investigation of interest is that of the transient response of a light bulb when switched on. The current through the bulb may be monitored by measuring the voltage across a series resistor. If the storage scope is triggered by this voltage, then it will be seen that the current is initially high and falls to its equilibrium value as the lamp warms up. This takes about 0·5 s for a 12 V 24 W filament bulb: use a heavy duty 1R resistor in series.

It is interesting to superimpose the graph of light intensity. The triggering is again provided by the voltage across the resistor. The light output may be observed using a photodiode and amplifier: one circuit for this was given in the article on scanning optical interference patterns in *Electronics and Computing Monthly,* April '83. (If you find the op amp circuit is unstable, connect a capacitor (say 100 n) between the V+ and V– supplies close to the i.c. **Fig 9** shows the sort of result that can be obtained.

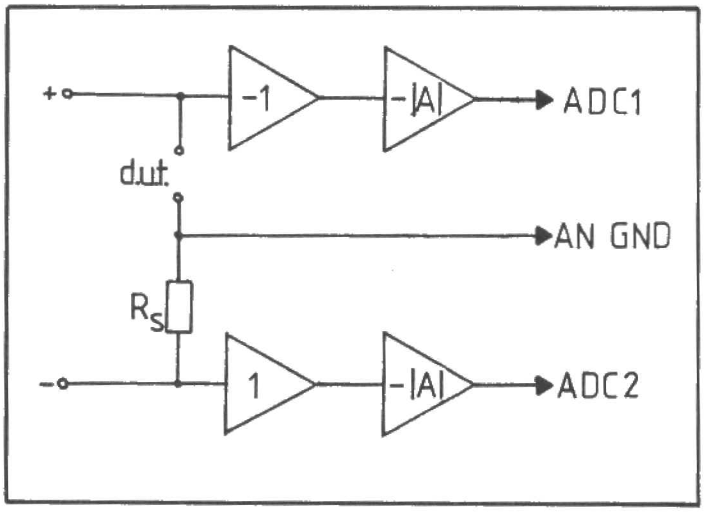


Figure 5. Measuring V and I using op amps to scale voltages. –1: inverting buffer. 1: non-inverting buffer. –A: inverting amplifier.

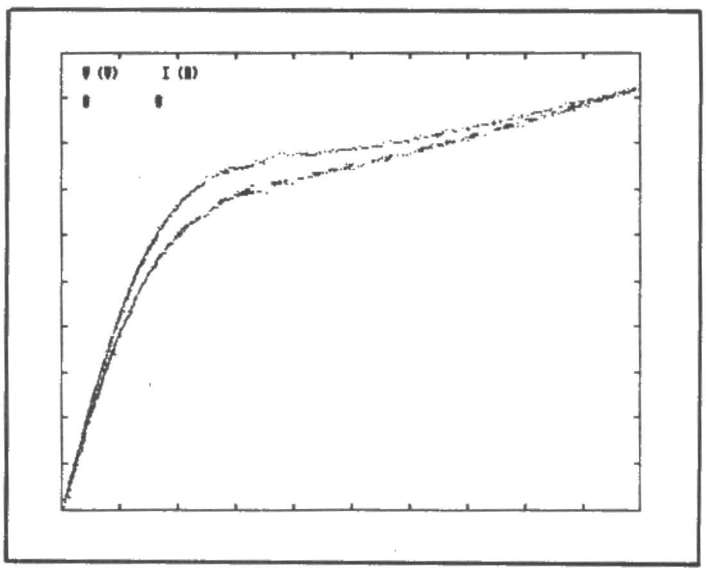
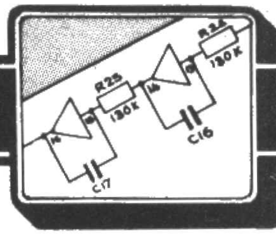


Figure 6. Current vs. voltage for a light bulb, heating and cooling.



A storage scope provides a useful way of displaying a.c. circuit demonstrations, for example the phase relationship between voltage and current in a reactive circuit. Slow (e.g. 0·1 Hz) a.c. demonstrations are effective, but sometimes the phase differences are not easy to see, either on a conventional CRO or on demonstration meters. A possible arrangement is shown in **Fig 10.** The storage scope ±5 V input is connected successively to points A and B to show the voltage across the capacitor and the current respectively. Unfortunately this gives a 180° phase shift of one of the signals. However, it is easy enough to invert one of the traces. In the STORCRO program (Dec. '82 *E&CM*) the values are stored as single bytes, in the memory locations 'data' to 'data' +255. So an "inverted trace" option could be included which replaced (?data) with (255 – ?data) etc. in PROCgraph. **Fig 11** shows the current in such a circuit leading the capacitor voltage (the larger amplitude trace) by 90°.

These are just a few ideas. The graphs shown in this article were screen dumps to an Epson printer, facilitating the production of handouts for analysis by students.

```
 10 REM***VIPLOT***
 20 MODE0
 30 @%=&208
 40 PROCstart
 50 PROCscreen
 60 REPEAT
 70   V%=ADVAL(1) DIV 16
 80   I%=(ADVAL(2)-ADVAL(1)) DIV 16
 90   V=V%*1.82/4095
100   I=I%*1.82/4095/Rs
110   PRINTTAB(2);V;TAB(12);I;
120   PLOT69,V%DIV4*factor*1279/1023,I%DIV4*factor
130   KEY$=INKEY$(0)
140   IF KEY$=" " PROCstart
150   UNTILO
160
170 DEF PROCstart
180 CLS
190 INPUT "Max voltage",VMAX
200 factor=1.82/VMAX
210 INPUT"Series resistor",Rs
220 ENDPROC
230
240 DEF PROCscreen
250 VDU23;8202;0;0;0;
260 CLS
270 VDU19;4;0;
280 DRAW0,1023:DRAW1279,1023:DRAW1279,0:DRAW0,0
290 VDU5
300 FOR Y=15 TO 1035 STEP 1020
310   FOR X=122 TO 1279 STEP 128
320     MOVE X,Y
330     PRINT"!"
340     NEXT
350   NEXT
360 FOR X=0 TO 1264 STEP 1264
370   FOR Y=118 TO 1023 STEP 102
380     MOVE X,Y
390     PRINT"-"
400     NEXT
410   NEXT
420 VDU4
430 PRINTTAB(3,1);"V (V)";TAB(14,1);"I (A)"
440 VDU28,1,3,25,3
450 ENDPROC
```

Program 1. Program for plotting V-I curves using two channels of the a-to-d converter.

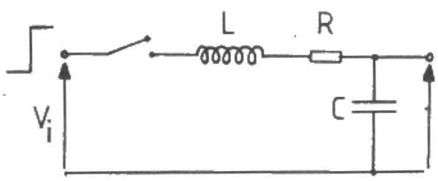


Figure 7. A step voltage applied to an LCR circuit.

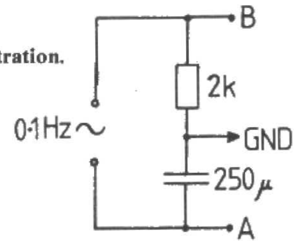


Figure 10. Low frequency AC demonstration.

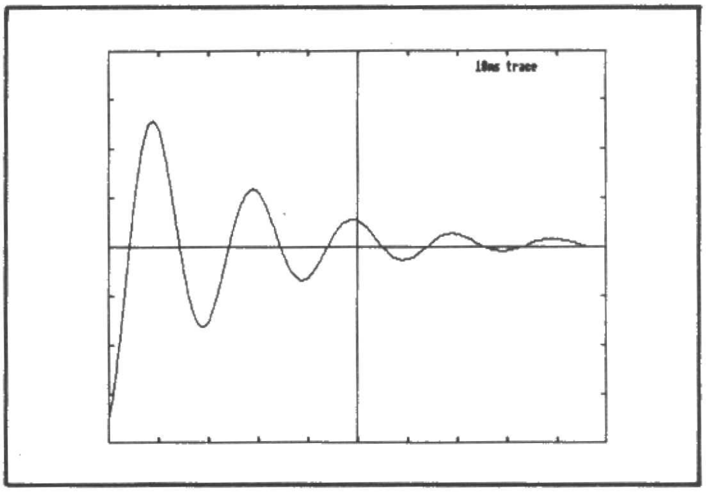


Figure 8. LCR transient.

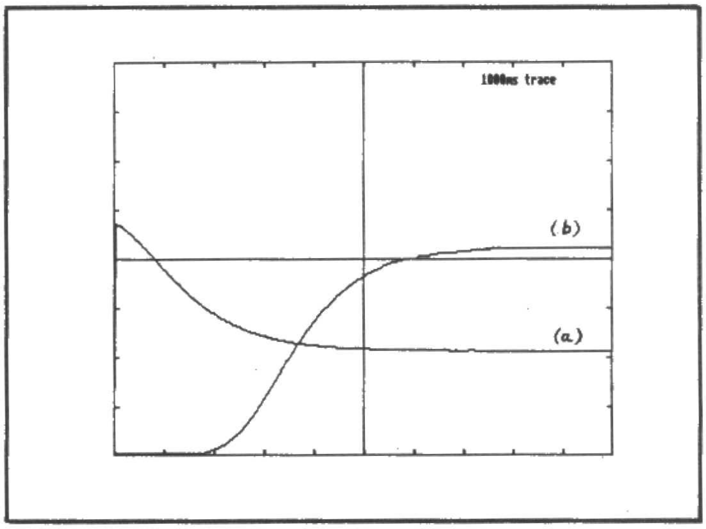


Figure 9. Graphs of (a) current, (b) light intensity, for a bulb when switched on.

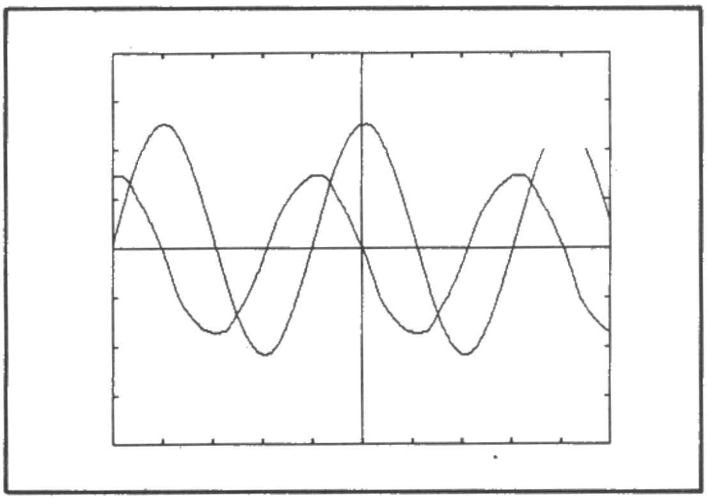
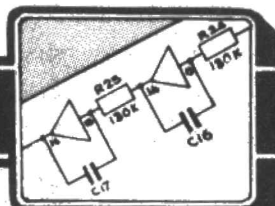


Figure 11. Voltage and current in a reactive circuit displayed using the BBC Micro as a storage scope.



POWER CONTROL FOR MICROS

Malcolm Plant describes three circuits that enable a micro to interface with power handling devices.

Three control units are described: a relay driver unit for on-off control of power to an electrical load; a digital-to-analogue converter (DAC) unit for smooth power control of lamps and DC motors; and a stepper motor controller unit for precise angular positioning of a stepper motor.

These control units have a number of useful applications. For example, the relay driver unit can be used for switching points and lamps and operating motors and controlling sound effects in model train systems or, at a higher power level, switching domestic lighting and in thermostat applications – for example, aquaria and horticulture. If you use the relay driver unit in association with the infra-red control system described in the March issue of *Electronics & Computing*, remote control of any of the above is possible. For example, the model train system, or any model for that matter, can be controlled from across the room using the hand-held transmitter unit. The Basic program is constructed so that it continually searches for significant four-bit codes on the read port of the in-out port which the program interprets to provide relay energising signals via the write port. And you might consider dedicating a ZX81 micro to the on-off control of a number of domestic appliances, TVs, hi-fi, room lights, for example.

The DAC unit is intended for the smooth speed control of motors or for controlling the brightness of filament lamps. It is designed to convert an 8-bit data word which the computer sends to the write port of the in-out port to an equivalent analogue voltage. The unit has the facility for unipolar or bipolar operation so that the DC motor can be controlled in forward only, or forward and reverse motion, respectively. That means that a model train, for example, can be moved to simulate real motion.

The stepper motor controller unit is designed to precisely position robot arms or steering mechanisms. Signals it receives from the computer's write port are decoded to rotate the stepper motor spindle through precise angles which are multiples of the motor's step angle. Clockwise or anticlockwise motion is possible merely by pulsing two bits of data at the write port in a logical sequence.

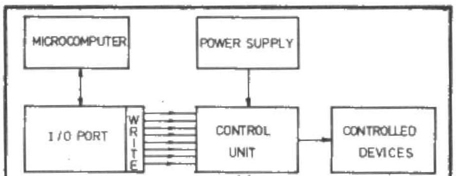


Figure 1. Block diagram of a typical power control system.

These three power control units have one electrical characteristic in common: they each act to boost the micro or milliampere signals, usually at 5V, to currents of a few amperes to hundreds of amperes, if required, in the controlled devices. **Fig 1** summarises the control system in relation to the micro's in-out port. The control units have been evaluated for the two Sinclair machines but are quite suitable for other micros which have appropriate in-out decoding circuits fitted to them.

Relay Unit

Generally, a Darlington pair transistor is used to amplify the current in the I/O port circuitry to a level suitable to energise a coil of an electromagnetic relay. Some designers use a field effect transistor, usually a VMOS type. The relay effectively isolates the I/O port, and thus the computer, from any high currents or high voltages which the relay contacts are called upon to switch on and off.

The circuit of **Fig 2** uses an integrated circuit, IC1, containing seven Darlington pair transistors. Only four of the transistors are used but it is possible to use all seven by a redesign of the layout of the relays on the circuit board. If you really want to control eight relays using the eight channels available from the micro, use an octal driver IC as shown in **Fig 3.**

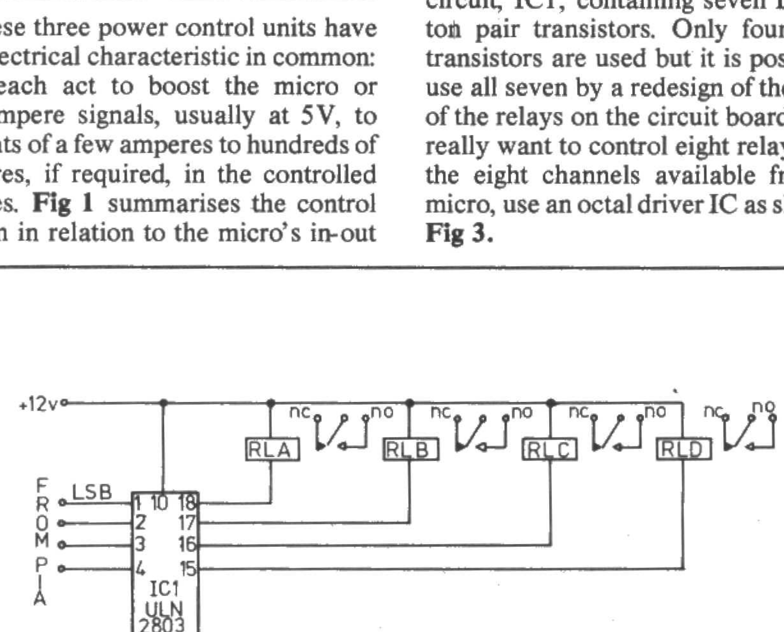


Figure 2. Circuit diagram of the relay unit.

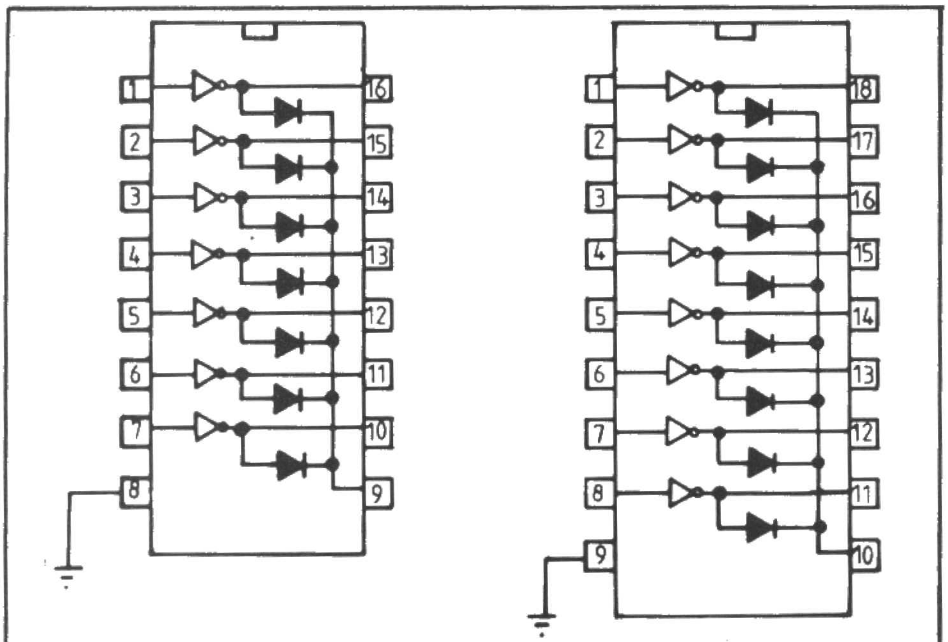


Figure 3. Pin-outs for two ICs providing Darlington pair power drivers. These IC's should be available from the component suppliers advertising in *E&CM.*

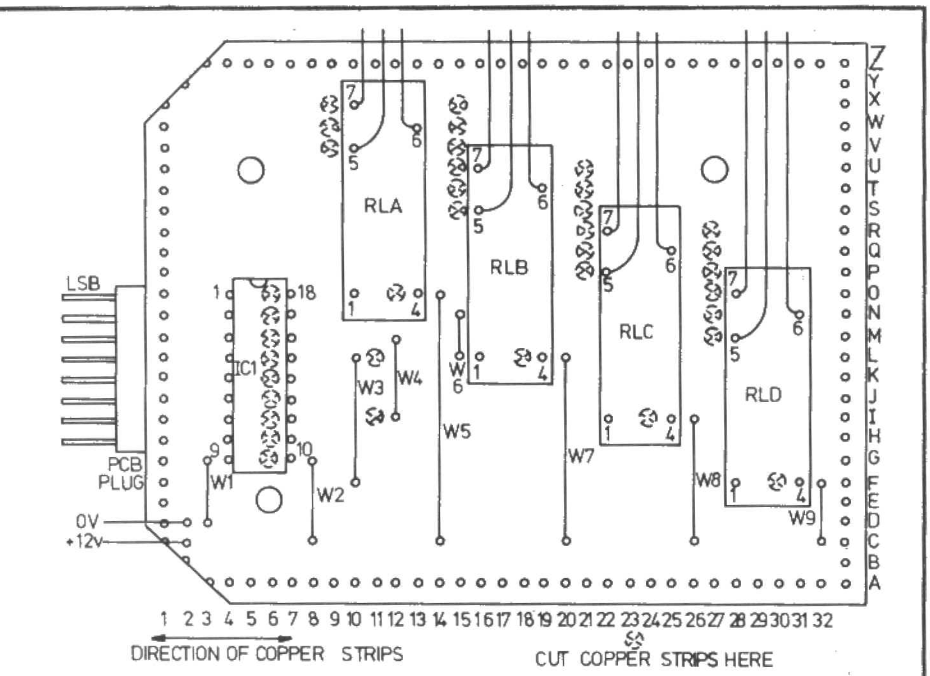


Figure 4. Stripboard layout of the Relay Unit.

Assembly

Figure 4 shows the stripboard layout for the relay driver unit. As the component count for this interface is relatively low, construction should not prove too onerous. The major points to note are the track cuts as indicated in **Fig 4.**

The stripboard should be placed in contact with the underside of the lid of the box and the three 3.3mm diameter fixing holes marked and drilled. Use nylon screws and nuts to fix the stripboard to underside of the lid using three of the nuts as spacers. Two of these screws should also pass through the terminal block fixed to the lid of the box.

Holes should be drilled close to each group of three terminal screws to take the three leads from each relay. Also drill two 8mm diameter holes for the 4mm plugs of the power supply at the end opposite the PCB plug. Fit the 4mm sockets and solder the power supply leads to them from the circuit board.

The four groups of three leads from the relay contacts should be passed through the four sets of holes in the lid of the box and secured to the terminal block. Tighten the circuit board and block using the three fixing screws. Cut a slot in the rim of the end wall of the box to accommodate the PCB plug. Mark the box to indicate the least significant bit corresponding to pin 1 of IC1. Finally insert IC1 into its socket and fit the lid of the box to its base.

The Relay Unit is tested by connecting a 'flying' PCB socket to the PCB plug mounted on the side of the unit and plugging in a 12V DC supply to the 4mm sockets on the unit. You must make sure that the 0V of the supply is common with the 0V supply of the I/O port. Now test the Relay unit by switching the appropriate port bits on and off.

Just a few words of warning. If you are contemplating switching mains voltages with the relays specified, be sure that you take full precautions. Note that the connecting wire specified can only carry 1.4A max although the relays carry more. Mains currents should not be conducted along the tracks of the stripboard. Earth a metal box, if used, to the earth lead of the mains.

PARTS LIST

Semiconductor

IC1 ULN2803A

Relays

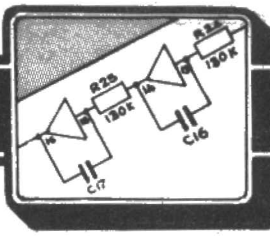
RLA - RLD 12V SPDT, 0.1 pitch

Miscellaneous

Veroboard (0.1 matrix, 26 x 36), PCB plug, 18 pin DIL socket, connecting wire, etc.

DAC Unit

It certainly isn't necessary to design a digital-to-analogue converter from scratch since there are many different types of ICs available to do the job



effortlessly. The circuit of **Fig 5** uses one of these, IC2 to provide an analogue equivalent of a byte of data from the write port of the computer. The analogue output from IC2 is passed to an op amp, IC3, which is wired as an adjustable gain noninverting voltage amplifier to give a symmetrical bipolar output voltage (—10V, 0V, +10V) if resistor R2 is included in the circuit by switch SW1. If SW1 is open, a unipolar output (0V, +10V) is obtained. A medium power output stage suitable for operating small DC motors and lamps is provided by the complementary Darlington pairs of transistors. Note that IC1 requires a single rail 5V power supply which is obtained from 12V by means of the voltage regulator IC1.

Assembly

Using 0.1 matrix stripboard as shown in **Fig 6** the assembly of the circuit is uncomplicated provided the usual care is taken with soldering.

The stripboard assembly should now be placed centrally in contact with the underside of the lid of the box, and two 3.3mm diameter fixing holes, as shown in **Fig 6,** marked and drilled in the lid. Using four nylon nuts and two screws, you should fix the circuit assembly to the lid using two of the nuts as spacers. Make sure that the PCB plug's pins project beyond the edge of the lid.

Complete the circuit assembly by drilling two 8mm diameter holes along one side of the stripboard and three along the other side. Fit the red and green power supply sockets in the set of three holes and the two 'load' sockets (yellow) in the two holes. Wire up to these sockets from the circuit board. Mount the toggle switch in a 6.2mm diameter hole in the lid of the box and complete connections to this switch.

Cut the two heatsinks to size and bolt them firmly to the two power transistors. Make sure the metal is clear of bare wires and components. Fit the lid to the base of the box and mark and cut a slot in the rim of the box to allow the PCB plug to fit tightly when the lid is closed. Do not screw the lid down until the output voltage has been set.

Connect the DAC to a ±12 supply and connect it to the I/O port using a 'flying' PCB plug. Make sure you have a common 0V connection between the 0V supply to the I/O port and the 0V (green socket) of the DAC. Plug in a 12V 2.2W

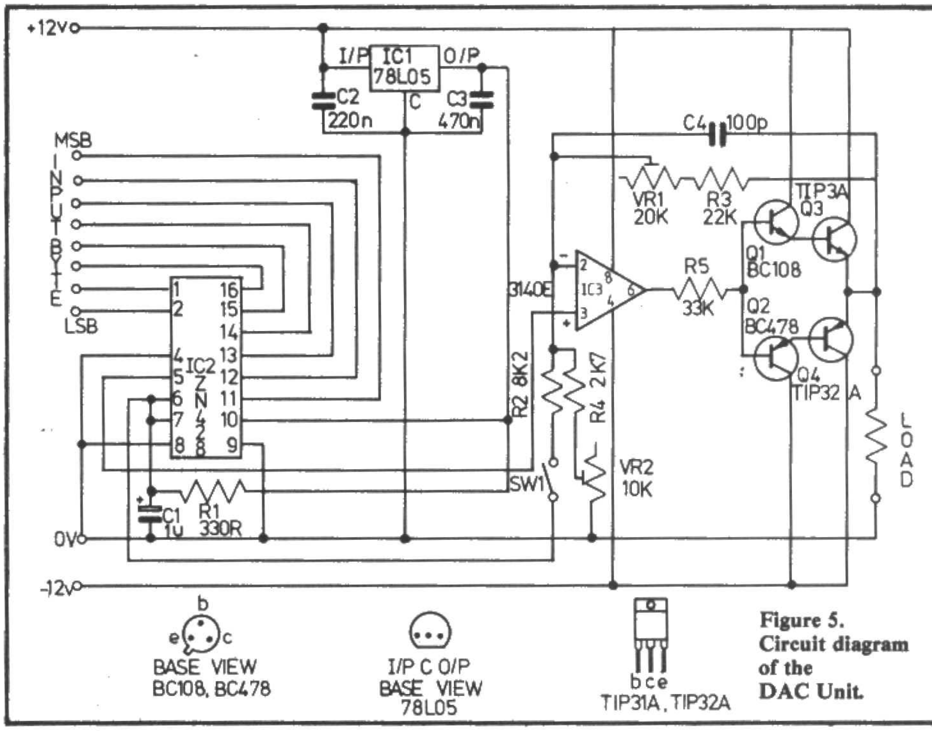


Figure 5. Circuit diagram of the DAC Unit.

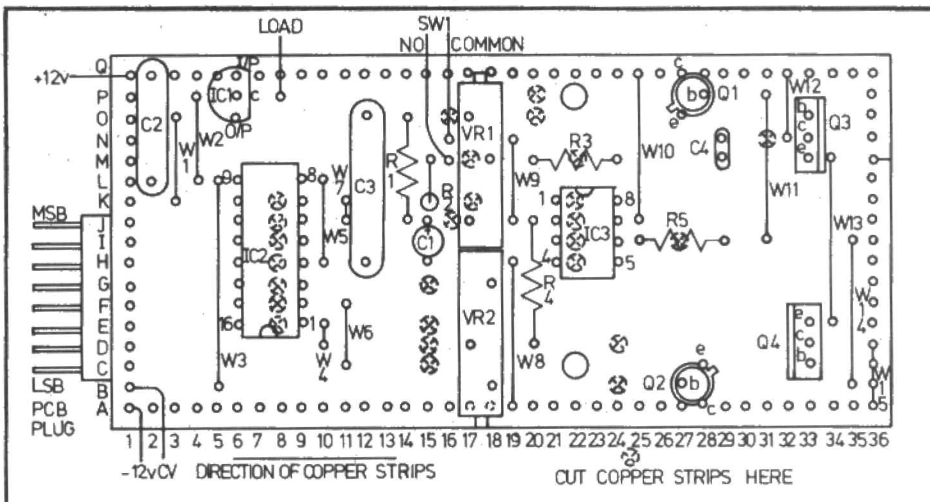


Figure 6. The DAC Unit's strip board layout.

PARTS LIST

Resistors (all ¼W 5%)

R1	390R
R2	8k2
R3	22k
R4	2k7
R5	33k

Potentiometers (20 turn cermet)

VR1	20k
VR2	10k

Capacitors

C1	1u0
C2	220n
C3	470n
C4	100p

Semiconductors

Q1	BC108
Q2	BC478
Q3	TIP31A
Q4	TIP32A
IC1	78L05
IC2	ZN428E
IC3	CA3140E MOS/FET

Switches

SW1 miniature toggle switch SPDT

Miscellaneous

Veroboard (0.1 matrix, 17 x 36), DIL sockets, PCB plug, connecting wire, etc.

lamp to the yellow sockets to act as a load and across this load connect a voltmeter to measure the DC voltage across it. Set SW1 to its open position so that the DAC is in the unipolar mode. Set all bits from the write port to zero and adjust VR1 so that the voltage reads zero. Now set all bits to 'high' and adjust VR2 so that the voltage across the load is, say, 6V. Program the write port to change its 8-bit value from 0 to 11111111. The lamp should a) brighten continuously from zero to full brightness when all bits are on if the switch is in the unipolar position and b) start bright and dim to zero (0V) for a binary value of 10000000 and brighten to maximum (6V) when the value is 11111111. A DC motor's speed can now be altered clockwise and anticlockwise if the DAC is in the bipolar mode and one direction only if in the unipolar mode.

Stepper Motor Unit

The SAA1027 is a purpose-designed IC for driving the phases of stepper motors in the correct sequence so that the axle of the stepper rotates by a precisely determined angle. The step angle depends on the design of the motor and the one used in this circuit is an Impex ID35 which is a four phase type having a step angle of 7.5 degrees and draws about 240mA per phase.

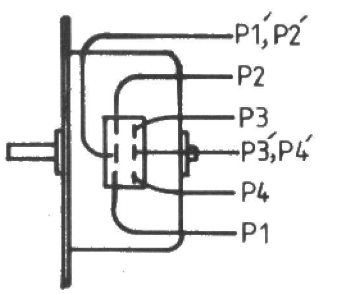
In this circuit both the SAA1027 and the motor are driven by a 12V DC supply. At this voltage, the pulse bit (pin 15) and the direction bit (pin 3) are logic HIGH for voltages between +7.5V and +12V and LOW for voltages between 0V and +4.5V. The necessary logic conversion from the 5V logic of the in-out port to the logic of IC3 is performed by IC1 which is a quad 2-input NAND gate. The pulse bit and the direction bit are first inverted by two of the NAND gates while the pull up resistors, R2 and R3, on the second pair of NAND gates provide the logic level conversion. IC1 provides the regulated 5V supply for IC2.

The stepper motor driver circuit requires the following signals from the in-out port:

pulse bit: each pulse must be at least 30ms long and each step angle of the motor begins on the positive going edge of the pulse

direction bit: a logic HIGH causes an anticlockwise rotation of the motor and a logic LOW a clockwise rotation viewed from the shaft of the stepper motor.

In this circuit, pin 2 of IC3 is set to logic HIGH permanently so as to enable phase switching to occur. If it is taken to logic LOW, phase switching is disabled.



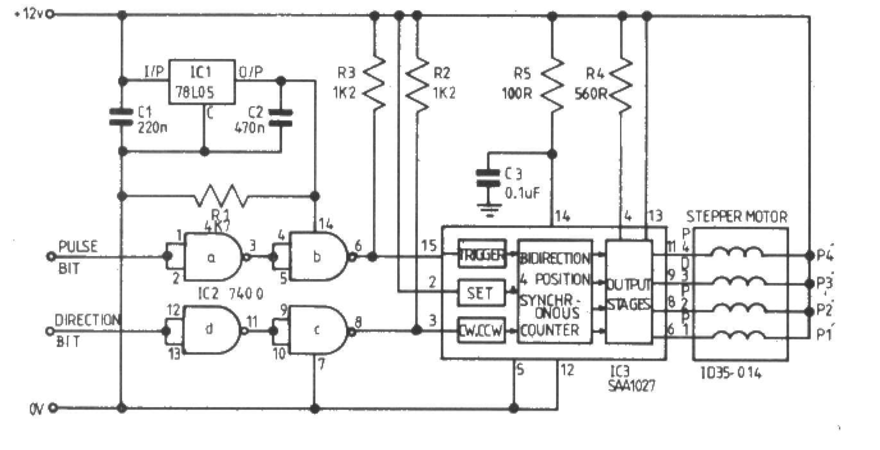


Figure 7. Circuit diagram of the Stepper Motor Unit plus connection details for the motor.

Assembly

The assembly of the components on 0.1 matrix stripboard is shown in **Fig 8.**

When the board is complete solder 7/0.2mm wires at the following points: P35 to connect to the +12V 4mm socket; D36 to connect to the 0V 4mm socket; N35 and O35 to connect to the DC motor 4mm sockets. Note that these DC motor sockets enable a DC motor to be controlled via the PCB plug connected to the circuit. That is, you may want to control a DC motor as well as the stepper motor and this can be done by plugging in the DAC output to the DC motor sockets on the stepper unit. A PCB socket plugged into the unit can then carry the wires for a DC motor as well as for the stepper motor.

Place the stripboard centrally in contact with the underside of the lid of the box and mark and drill the 3 3.3mm diameter fixing holes. Use the three nylon screws and six nuts to fix the circuit board to the lid of the box using three nuts as spacers.

Drill four 8mm diameter holes in the lid of the box, two on each side of the circuit board for the power supply and DC motor 4mm sockets. Insert and fix these sockets and solder leads to them from the circuit board.

Cut a slot in each end wall of the base of the box to allow the two PCB plugs to rest in them when the lid is fixed to the base. Label the pins of these plugs as follows: 'pulse bit', 'direction bit', '+12V', 'DC motor', 'P1, P2, P3, P4'

PARTS LIST

Resistors (¼W 5%)

R1	4k7
R2, R3	1k2
R4	560R
R5	100R

Capacitors

C1	220n
C2	470n
C3	100n

Semiconductors

IC1	78L05
IC2	7400
IC3	SAA1027

Miscellaneous

Veroboard (0.1 matrix, 17 x 36) DIL sockets, connecting wire, PCB plugs, stepper motor, etc.

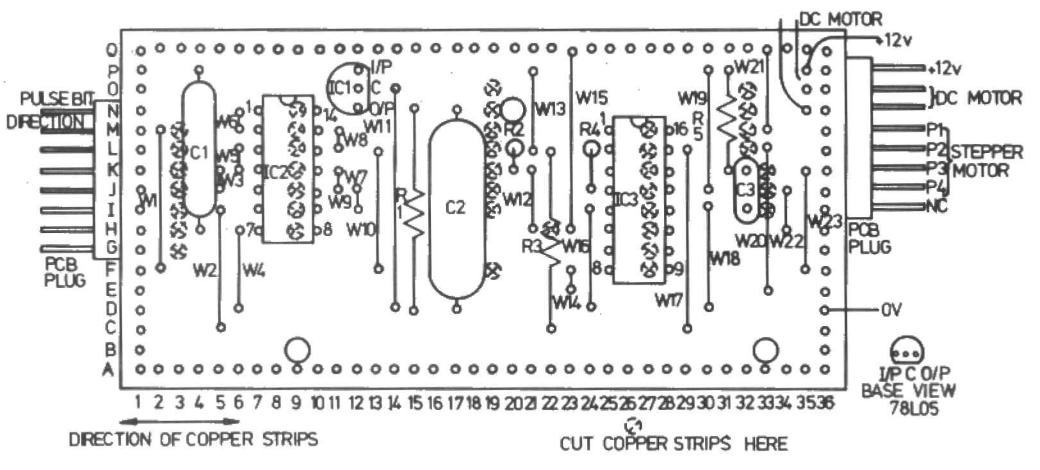


Figure 8. Stripboard layout for the Stepper Motor Unit.

for the four phases of the stepper motor.

Make up a flying lead which has a PCB plug on one end and connect this lead to the stepper motor as shown in the figure. Note that four of the terminals to the stepper motor are connected together to the +12V supply lead. Insert the ICs into their sockets and fix the lid to the box.

The Stepper Motor Unit is easily tested in Basic by switching on and off the first two least significant bits on the write port of the in-out port.

Next month we hope to publish PCB patterns for these designs. E&CM

Understanding Digital Electronics

Shift registers and counters are closely examined by J. Oliver Linton in Part Four of the series.

The most popular and versatile of all the latches is probably the J-K flip-flop. Its circuit and symbol are shown in **Fig 1.** It consists of a clocked $\overline{S}$-$\overline{R}$ master flip-flop followed by a slave D-type latch whose clock is inverted. The outputs of this latch are cross-coupled to the inputs of the master latch through two NAND gates whose other inputs are always known as the J and K inputs. The most important difference between the J-K flip-flop and the D-type latch is that it is pulse-triggered rather than edge triggered. This means that the data lines are read on the rising edge of a clock pulse, but only transferred to the output on the falling edge of the clock pulse. Actually, it is not quite true to say that the data is read on the rising edge because if the data is changed while the clock pulse is high, the output is not always as expected. For this reason, it is assumed that the data lines will be stable for the whole period of the clock pulse.

The other major difference is that the J-K flip-flop has three distinct modes of operation. If J and K are both LOW, the outputs remain unchanged whatever state they are in. If J is the inverse of K, then the device becomes a pulse-triggered D-type latch, the outputs of which become equal to the inputs on the falling edge of the clock pulse. Finally if J and K are both HIGH the outputs 'toggle' (i.e. change state) on the falling edge of every clock pulse. These modes are summarised in **Table 1.**

J	K	Outputs
LOW	LOW	Do not change
LOW	HIGH	Become equal to J and K at end of next clock pulse
HIGH	LOW	
HIGH	HIGH	Toggle (change state) at end of each clock pulse

A close look at **Fig 1** will show why this is so. If J and K are both LOW, the two NAND gates are disabled and their outputs will both be HIGH. The $\overline{S}$-$\overline{R}$ latch is therefore also disabled and will never change state. Suppose, however, both J and K are HIGH. If $\overline{Q}$ is HIGH and (Q is LOW) then J's NAND gate is enabled and the $\overline{S}$ input of the master latch is LOW. On the rising edge of the clock pulse, the Q output of the master latch will be SET (i.e. go HIGH) and on the falling edge, this output will be transferred through to the slave outputs. On the next clock pulse, it will be the K's NAND gate that is enabled so the outputs will change back again. This is what is meant by 'toggling'. In the intermediate mode in which either J or K is HIGH, only one of the two NAND gates is enabled and the latch will either toggle into a state in which the output Q equals J, or if it is already in that state, it will stay there.

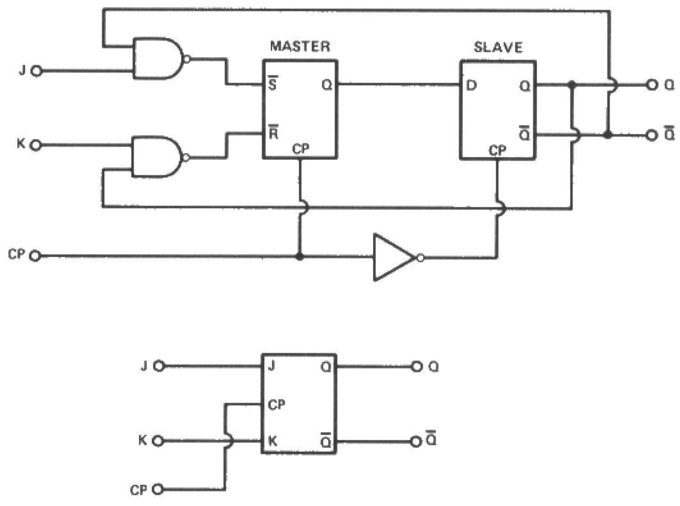


Figure 1. Block diagram and functional symbol of a J-K flip-flop.

CLOSER INVESTIGATION

It is well worth investing in a couple of dual J-K flip-flops in order to become familiar with these modes of operation and to try out some of the circuits to be described this month. The 7473 is the usual choice; the 7476 has additional $\overline{S}$ pins which SET the outputs when taken LOW. Both chips have $\overline{R}$ pins which RESET the outputs when taken LOW. It is worth remembering that with TTL logic, unconnected inputs will act as if they are HIGH (though it is bad practice to leave them floating in a permanent circuit). The R pin may therefore be left unconnected and if a toggling action is required, so may the J and K pins. Remember also to use a switch debounce circuit (described last month) on the clock input otherwise you will have trouble with spurious pulses.

The first real circuit we shall look at is the shift register shown in **Fig 2.** The Q and $\overline{Q}$ outputs of each flip-flop are directly connected to the J and K inputs of the next. Since the K input is always the inverse of the J input (including the first which is connected via an inverter) all the flip-flops are working in the intermediate mode. At every clock pulse, data from the previous latch (on the left) is read through to the output, the result being that any pattern of bits held in the register is shifted one place to the right. At the same time, one new bit is added to the left hand end and one bit 'falls out' of the right hand end.

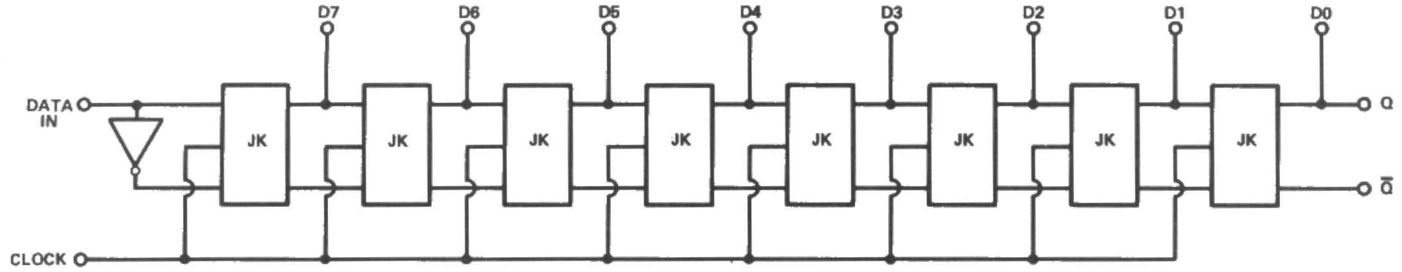


Figure 2. A series of interconnected flip-flops forming a shift register.

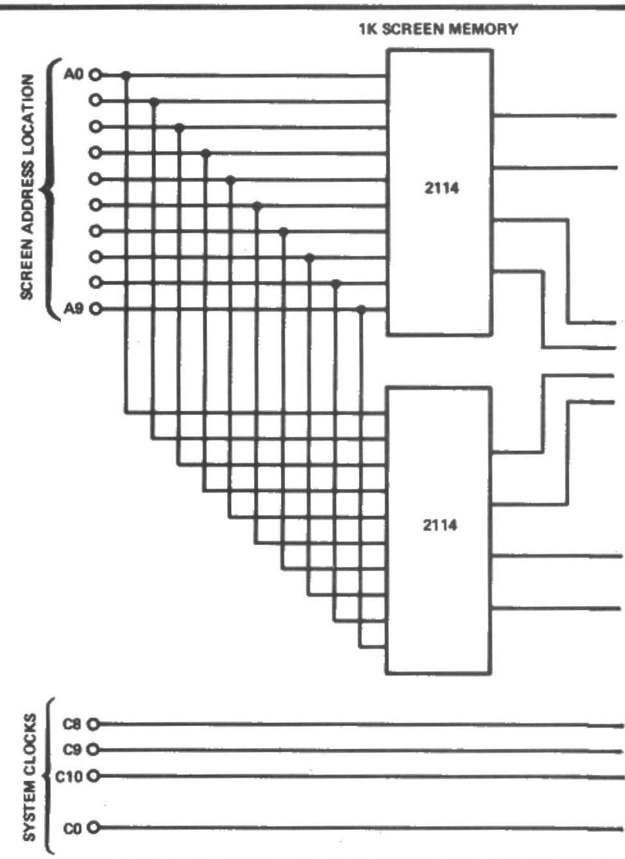
This arrangement is frequently used to convert serial data (e.g. from a cassette recorder) into parallel data suitable for input into a microprocessor. In a typical system data is stored on tape in the form of eight bits preceded and followed by one start and one stop bit. As each bit arrives, it is pushed into the register from the left hand side and when all eight bits have been received, the data on the lines D0 - D7 is transferred to the main data bus. A suitable chip to use in this application would be 74164 which is an 8-bit serial in/parallel out shift register. It has two data inputs which must both be HIGH for data to be read into the register and a master reset (MR) pin. In this device, the data is read on the rising edge of the clock pulse, not the falling edge.

In practice, the cassette ports of a modern micro are more usually handled by a sophisticated chip called a UART (Universal Asynchronous Receiver/Transmitter) which contains not only the shift registers required, but also the necessary circuitry to handle the timing and control signals that are inevitably needed as well.

Parallel To Serial

The process of converting parallel data into serial data is just as easy but requires a flip-flop with SET and RESET lines. First the register is cleared using the RESET line, then the data is loaded in parallel by means of the SET lines. (This is known as asynchronous or 'jam' loading because it is independent of the clock. A few shift registers have synchronous loading e.g. the 74166). Once the data has been read into the registers, it is shifted out of the right hand end of the register in series by applying the appropriate number of clock pulses.

The parallel in/serial out shift register can be used to write data to a cassette recorder or to a printer with a serial interface. it is also used to drive a Visual Display Unit in the following way. Most micros have a TEXT mode in which the screen is divided into about 1000 locations each of which can contain one character. The content of each location is held in a memory somewhere which contains an 8-bit number – thus allowing the generation of one of 256 different characters. As the electron beam scans across the screen, the appropriate memory location is addressed, but the 8-bit number found there only tells the computer which character is to be printed, it does not give any information about the shape of the character or where to put the dots on the screen. This information is contained in another memory (which may be in the computers main memory or it may be in a special Read Only Memory or ROM called a Character Generator). Assuming that the character is formed on an 8 x 8 matrix, eight 8-bit numbers are sufficient to define it. The **data** from the main memory location is used as part of the **address** which is applied to the character generator; in addition we need three further lines, derived from the system clock, to tell us which horizontal line of the character we are drawing. These eleven lines uniquely decode the appropriate 8 bits in the character generator which are read in parallel into a parallel in/serial out shift register at the start of every character. As the electron beam scans across the screen these bits are read out in series and the data is used to control the brightness of the spot of light created thus building up the shape of the character line by line. It is interesting to calculate that with 625 lines every 1/25 s, and 40 characters on a line each necessitating 8 bits, the data transfer rate is 4.8 Mbits/s. As you can see, these registers can really shift! (see **Fig 3**).



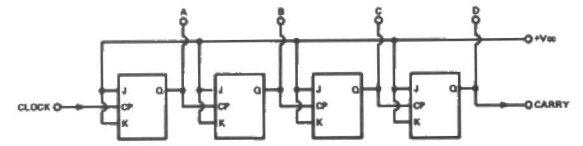
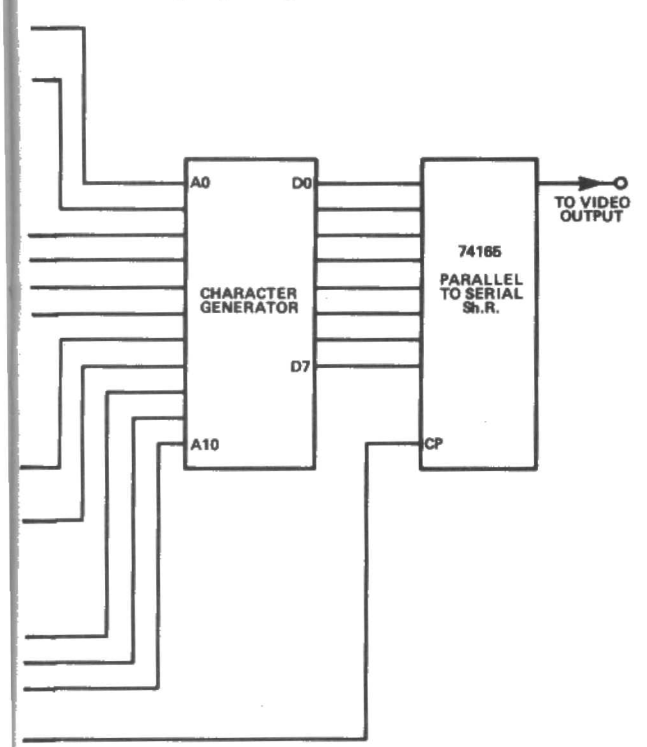


Figure 4. An alternative J-K flip-flop connection arrangement forms a binary counter.

Variation On A Theme

Figure 4 shows another way of stringing J-K flip-flops together. Can you figure out what will happen when a train of pulses is applied to the clock input? You will notice that the J and K inputs are simply held HIGH all the time. This means that the flip-flops will toggle every time they receive a **falling** edge on the CP pin. The cunning thing is that, unlike the previous case where all the clock pins were tied together, the CP pin of each flip-flop is connected to the **output** of the previous stage. In consequence, each flip-flop runs at exactly half the frequency of the previous one. Moreover, the four outputs A, B C and D count up through a binary sequence, A being the **least** significant bit (LSB) and D being the **most** significant bit (MSB). (It is a source of much confusion that in this circuit the least significant bit is at the **left** hand end!). It is now possible to connect the four outputs to a 7447 decoder/driver and seven-segment display and you have got an event counter. (Remember to use a switch debounce circuit on your CLOCK input though, otherwise a lot of spurious pulses will be counted). I will leave the reader to discover what happens if the CP pins of each flip-flop are connected to the $\overline{Q}$ output of the previous one, instead of the Q output.

Figure 3. Simplified form of a VDU circuit showing the use of a flip-flop as a parallel to serial converter.



Decimal Counters

The counter described above will count to 15 and then start again at zero. Suppose we want to make a decimal counter which resets to zero on the **tenth** clock pulse. At this instant the counter will read ten (1010 in binary) so outputs B and D will be HIGH. (Remember, the outputs appear to be the wrong way round in the diagram). Adding a NAND gate between the common reset line ($\overline{R}$) and the B and D outputs as shown in **Fig 5** achieves the desired result. As soon as these outputs go HIGH, the reset line goes LOW resetting all the flip-flops back to zero. In most applications, the brief '10' output does not matter. If you want to make a multi-digit counter, the last digit of one 4-bit counter can be used to clock the next one.

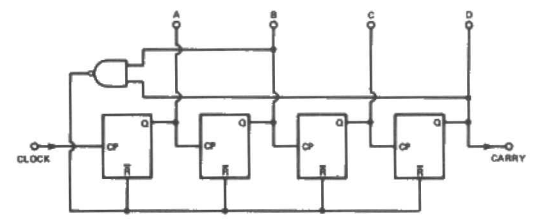


Figure 5. By resetting a binary counter to zero on the tenth count, a decimal counter can be implemented.

By gating other outputs in various ways to the reset line, the counter may be made to reset at any desired number. In this way it is possible to divide any frequency by any whole number – for example, the TV line and field frequencies can be obtained from a frequency of 31250Hz by dividing by 2 and 625 respectively.

In order to make a stop watch or clock, the tens of seconds counter and the tens of minutes counter must rest on 6. Such a counter can be made from three J-K flip-flops and one NAND gate. All the necessary components for this and the decade counter will be found in a neat package – the 7490 whose configuration is shown in **Fig 6.** When used as a decade counter, the output from the first flip-flop must be connected to the CLOCK1 input (i.e. pin 12 must be connected to pin 1).

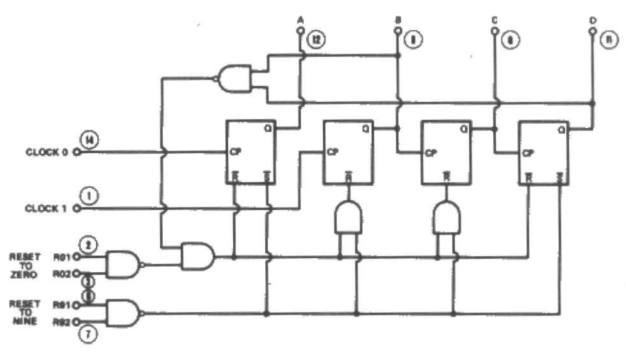


Figure 6. The internal arrangement of the versatile 7490 decade counter.

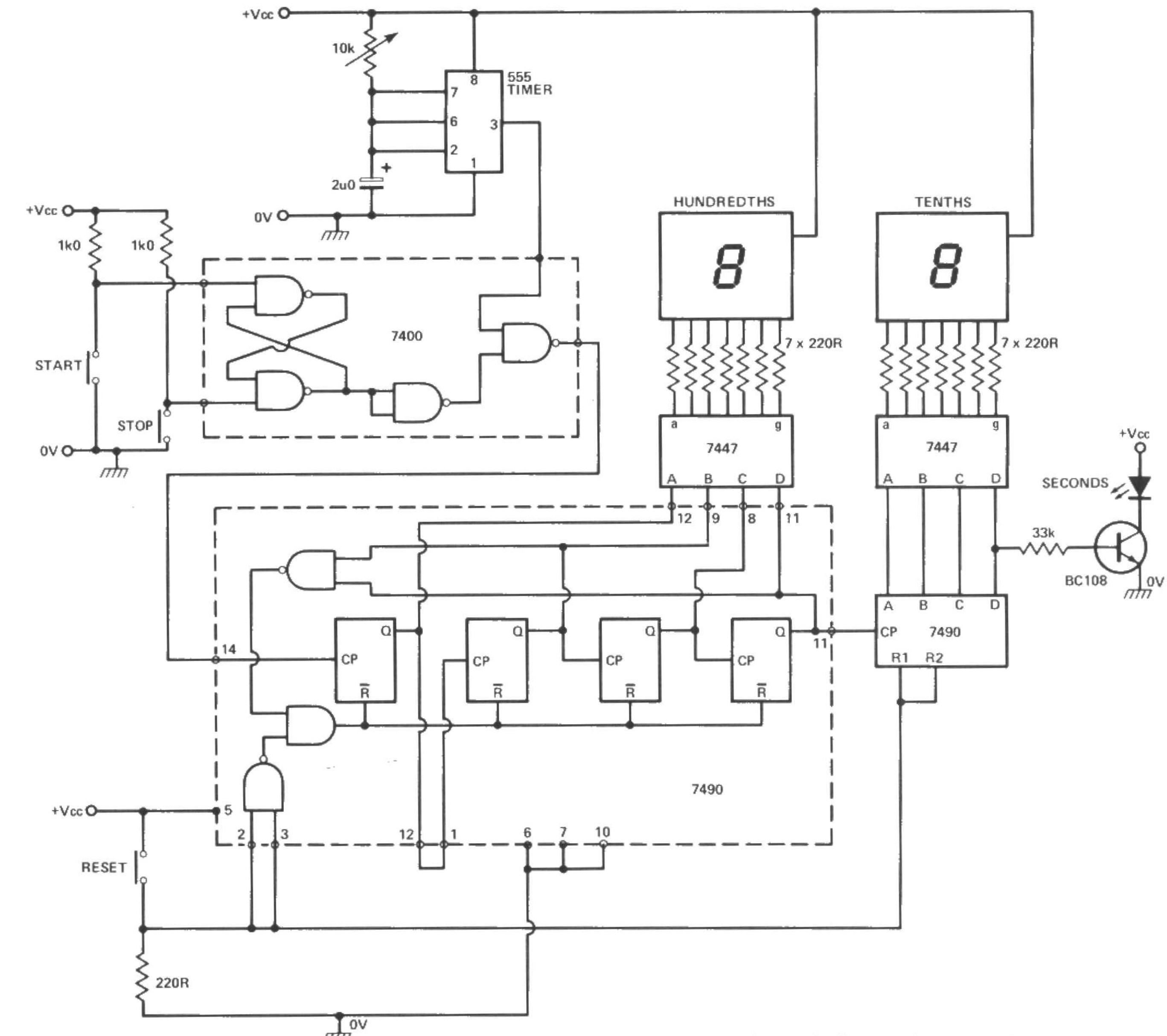


Figure 7. Circuit diagram of a reaction timer that makes use of the many building blocks so far described in this series.

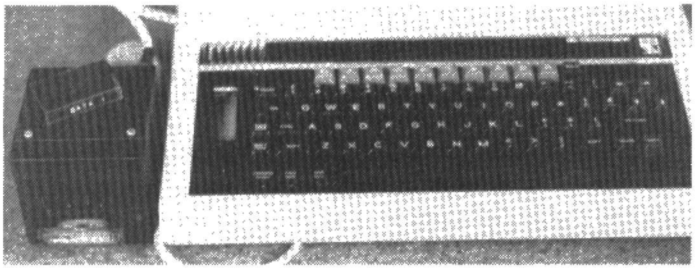
Notice also that if the 'reset to zero' and 'reset to nine' outputs are not being used, they should be grounded because if they are left floating they will continually reset the device. The reason for separating the first flip-flop from the other three is to allow the chip to produce a divide-by-ten function with an output square wave with an equal mark-space ratio. This is done by using the B, C and D flip-flops **first** as a divide-by-five counter and connecting the D output to CLOCK0. The output at A will be divided by ten but since the last stage was a divide-by-two stage, it will have a symmetrical form. The 'reset to nine' inputs are used in one form of subtraction known as 'nine's complement'. The 7493 is very similar to the 7490 but is a straightforward 4-bit counter with two 'reset to zero' inputs, and the 7492 is a divide-by-twelve counter.

And now to finish, **Fig 7** shows a complete circuit for a simple reaction timer. If you have been following this series, it should be clear exactly how it works except for the 555 timer which is used to provide negative-going clock pulses at a frequency of 100Hz. This clock can be calibrated by adjusting the 10k preset resistor so that the LED flashes exactly once every second. My reaction time is around 0.17 s. Is yours any better? If it is, have a few drinks to celebrate – and then try again!

E&CM

The Ikon HOBBIT

David Green BSc looks at a useful mass storage system for the BBC computer.



With the growth in popularity of the BBC micro as one of the few machines meeting the needs of the serious programmer, it's surprising that the choice of storage media has been confined to either an expensive floppy disk system or the infamously unreliable cassette. The alternatives, however, are about to be expanded following the introduction, by Ikon Computer Products, of a digital cassette recorder and interface supporting the full set of disk commands.

The new Ikon tape machine, called rather amusingly The Hobbit – a reference, no doubt, to its tireless wanderings up and down the tape – comes complete with ROM chip, connecting leads and mini cassette (specially designed for digital recording; though the standard audio product is also compatible). The cassette drive, made by Philips, is housed in a cube-shaped box with four chunky rubber feet on the base. These turned out to be invaluable, since the ribbon cable linking the Hobbit and BBC micro is barely long enough to allow the mechanism to sit on the top of the computer. Matters were further complicated when it came to connecting the power cable, which is terminated by a square matrix plug for the BBC's 'Power Out' socket. Upon attempting to insert the plug, no socket was to be found, due to the fact that some BBC computers possess a fully enclosed PSU. One quick call to Carmarthen (Wales), revealed that the required supply is +12V to the brown lead and 0V to the green/yellow (the blue is not needed) wire – under the circumstances, perhaps mains cable was not a particularly good choice!

ROM For Expansion

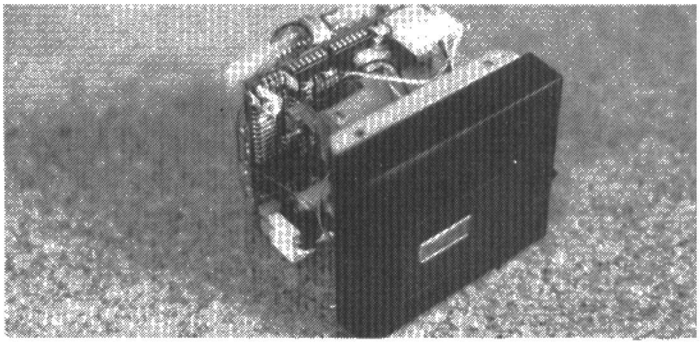
An important part of the Hobbit system is the EPROM IC, which must be installed in one of the BBC's expansion sockets. The IC – a 2764 UV blown erasable programmable ROM – can be positioned in any of these sockets (IC88, 100 or 101), thus leaving any other ROMs (e.g. a wordprocessing package) physically unaffected.

The procedure for setting up the Hobbit's software-derived OS is not as straightforward as the manual would suggest. This is partly due to the inadequacies of that tome, and partly to the variations on a theme of 'disassemble, plug-in, swap over, re-assemble and switch on'. Unscrewing the Beeb is no great task, but locating the various components outlined on page one of the manual is tedious. Then comes the tricky bit – inserting the ROM chip without breaking any pins or missing the holes altogether. Fortunately, the author has had some experience with similar devices, but for those of a less confident disposition, Ikon offer an installation service for £5.75 inclusive.

Once the IC has been installed, the link/diode corrections made and the BBC screwed back together, it only remains for the ribbon cable to be plugged into the user port and the power switched on. You'll soon know if everything is OK, since the display shows (amongst other things) Ikon's telephone number – taking calls, incidentally, seven days a week.

Handling The Hobbit

With everything up and running, the first command (with a new cassette) is to carry out a formatting routine. This seemed to take ages, with the added inconvenience of being unable to utilise the normal keyboard functions – a result of bad design on the part of Acorn rather than Ikon, the latter being presented with the choice of either catering for all versions of the BBC's operating systems, or confining their unit's compatibility to certain machines only. Both sides of the tape require a unique name, enabling the Hobbit to tell whether it has processed a particular side or not. The author recommends that both sides are catalogued (using *CAT) before being written to, preventing a 'TAPE FULL' message appearing as a result of bad housekeeping. In addition, the *CAT command provides useful information regarding volume name, file name, type of file and its size.



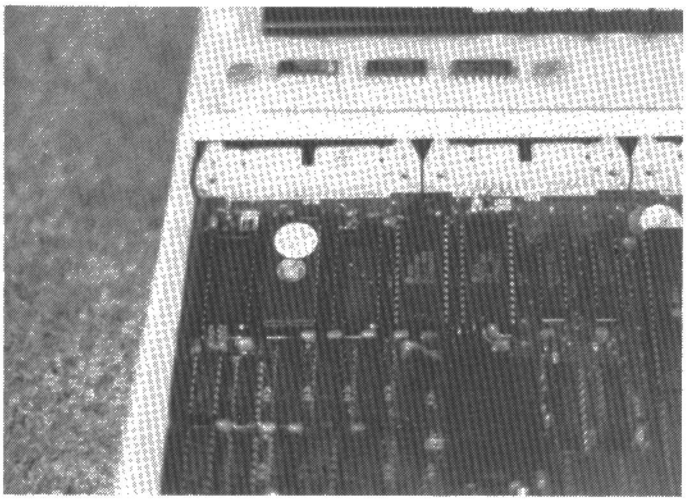
An internal view of the Hobbit revealing the minimal amount of electronics associated with the system.

After these preliminaries have been dealt with, the Hobbit can be treated as a cassette under keyboard/program control – no need to 'PRESS RECORD and return', it's all done for you. To save a program, just type the usual SAVE command, which will automatically power-up the cassette. Depending on the state of the tape – the number and proximity of blocks of bytes constituting other files – at the time when a read or write operation is desired, the delay before control returns to the keyboard can be as much as 1½ minutes; though somewhere between 15 and 35 seconds is more usual. Although direct

comparison against standard cassette recorders is inappropriate, since the Hobbit is about five times more expensive (at £155.25), it's interesting to note that the latter is able to read/write up to seven times faster. The reason for this lies in the fact that the Hobbit runs the tape at 7½ ips, permitting data transfer at 750 bytes (characters) per second (a typical figure for domestic recorders is 120 bps).

A Commanding Position

Aside from supporting the regular file handling commands, such as LOAD, SAVE and RUN, the software provides a number of novel capabilities. For instance, should you accidentally delete a file and then wish to restore it, the command *RECOUP can be used (assuming another file has not been created between DELETE and RECOUP). This is possible because the method by which space is allocated utilises a 'tagging' system, whereby active files are marked with pointers for as long as that space is not required for a new file.



The Hobbit's ROM fits into one of the BBC's vacant sockets.

The technique of marking data blocks during movements of the tape past the head, makes provision for random access files – information is not stored in one continuous chunk, as with conventional cassette systems, but as a series of blocks spread along the length of the tape. Data being transferred is first stored in a buffer, before being recorded, and the SAVE acknowledged – control is then returned to the keyboard. As mentioned earlier, this 'tying-up' of the BBC's command monitor is slightly irritating, especially when it is realised that once the tape buffer has been filled, other operations are carried out by the Hobbit automatically.

The loading procedure is exactly the reverse of the above, with the information being retrieved one block at a time and stored in a second buffer before entering the BBC's resident RAM. During both SAVE and LOAD operations, the Hobbit makes use of a central directory or catalogue of files. Hence, when a file is accessed, the tape drive is manipulated so that the head starts at the beginning of the directory. The file is then located in the following section and the reference marker used as a pointer to the first data block. Successive pointers then take the tape head to each block comprising the file – the difference between loading and saving depends on whether a marker is laid down or picked up.

The complete list of commands supported by the Hobbit is shown in **Table 1.** These replicate the disk commands, with a few unique additions. One important extra (formerly *BBC), allows control to be returned to the normal BBC OS. This is accomplished by holding down the 'O' key and pressing BREAK, then typing 'OLD'.

TABLE 1

*CAT	provides file information from directory
*COPY	copies one file to another
*DELETE	deletes files (individually)
*EXEC	accepts file lines as if coming from the keyboard
*FORMAT	identification and initialisation
*HOBBIT	returns to Hobbit OS from standard cassette system
*KILL	deletes all files on the tape
*LOAD	loads a machine code program
*RECOUP	restores a deleted file provided no other files have been created subsequently
*RENAME	changes a filename
*RUN	loads and runs object code
*SAVE	puts an area from memory into the named file
*SPOOL	puts the screen output into a file
*TAPE	returns to the BBC's cassette OS

Multiple Chains

A particularly useful feature of the Hobbit OS, is its active file capacity – the number of files which can be active at any one time – of five files. Each active file requires a separate buffer area (POKE addresses are given in the manual) of 750 bytes, but as soon as this has been allocated the file can be accessed in the usual way. So, for instance, if you're processing a set of data to be divided into certain subsets (e.g. age groups), the Hobbit will be able to cope with the various data transfers, without the need for program restructuring for serial storage methods.

Another advantage of the multiple file access facility is during program development. Often, having written a large program, specific sections will be applicable to others. Several of these commonly-used sections can be merged via the copy command to form a new program – imagine trying to do that on a standard cassette system.

These are just a few of the plus-points that the Hobbit scores over the conventional cassette. The comparison is, perhaps, unfair due to their differences in speed, means of storing information . . . and price. The same argument, however, could apply equally to the comparison with disk systems. What can be said is that the Hobbit makes an ideal complement to the BBC micro. It is many times more reliable and much faster than the usual cassette systems, whilst supporting the full range of disk commands – all in all a well designed package that should suit the needs of the vast majority of BBC micro owners.

E&CM

COMPUTER MODELLING

Brian Horsfield examines the 'Wankel' rotary combustion engine and describes a ZX81 program to model its action.

In theory a rotary engine is a much better proposition than a reciprocating engine since the purpose in both is to rotate a shaft. The German engineer, Dr. Felix Wankel developed the most successful rotary combustion engine. Wankel's rotary engine has only two principal moving parts, the Rotor and the shaft. As it rotates the rotor controls two ports, one for the intake of fresh fuel and the other for the removal of exhaust gases. The triangular shape of the rotor divides the (almost) figure of eight shape of the engine interior into three chambers. The turning rotor is continually increasing and decreasing the size of these three 'moving chambers', each one being an analogue of the cylinder in a standard piston engine.

As the rotor rotates, the volume of one chamber becomes smaller in a compression phase. When the volume is a minimum the spark plug fires to initiate the combustion phase. The volume of that chamber then increases up to the point where the exhaust port is uncovered. Its volume then decreases to chase the spent fuel out of the chamber. As soon as the exhaust phase is completed and the exhaust gases have been expelled from the chamber then that chamber moves over the inlet port and its volume increases once more so that fresh fuel mixture can be taken in before the next compression phase begins.

The three chambers are always at different stages of the four stroke cycle and each of the three chambers completes a full cycle of changes while the rotor turns through one 360° revolution. While one chamber may be on the intake phase, the compression and the ignition phases take place in another chamber and the third chamber will be on the exhaust phase.

The rotor moves in a circular orbit around the output shaft and as a consequence each complete revolution of the rotor results in three revolutions of the output shaft.

Minor Problems

Rotary engines suffer from a tendency to leave some of the fuel mixture unburned. This reduces the engine efficiency and also produces an undesirable source of atmospheric pollution. Another problem is concerned with the need to ensure that the three combustion chambers remain gas tight. Seals at the rotor tips and faces can largely overcome these problems.

The Wankel Rotary engine has several advantages. It has fewer moving parts and is lighter and more compact than comparable piston engines and has a superior power to weight ratio. The rotor and the output shaft both rotate in the same direction and at constant speeds. This has the effect of reducing engine vibrations and also makes the Wankel engine markedly quieter than a reciprocating engine. The Japanese automobile manufacturer, Mazda, has used Wankel engines in some of its vehicles over the past few years.

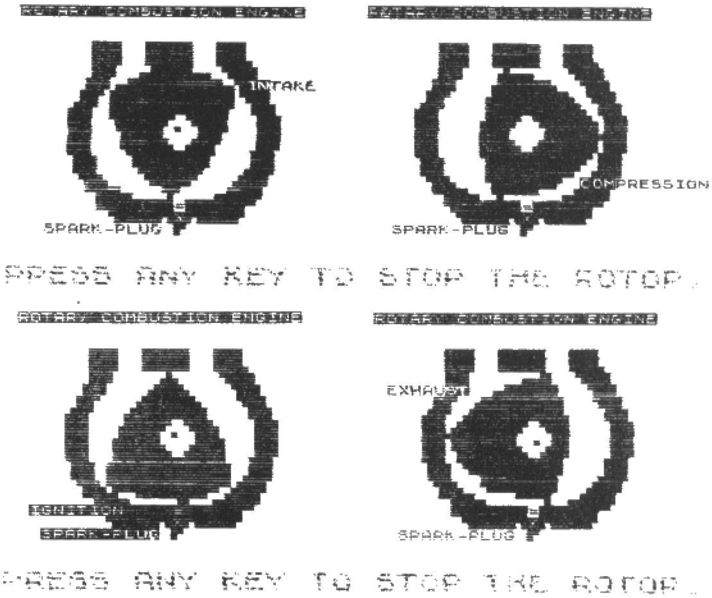
```
   0 REM (C)B.HORSFIELD JAN.83
  10 PRINT TAB 3;"                              "
  20 GOSUB 1000
  30 LET X=1
 100 FOR F=1 TO 4
 110 IF INKEY$<>"" THEN GOTO 110
 120 PRINT AT 5,0;P$(F)
 122 IF F=1 THEN PRINT AT 6,22;"  TAKE"
 124 IF F=2 THEN PRINT AT 14,21;"   PRESSION"
 126 IF F=3 THEN PRINT AT 16,4;"           ";AT 16,16;" ";AT 18,5;"           "
 128 IF F=4 THEN PRINT AT 16,4;"      ";AT 16,16;"=";AT 18,5;"SPARK-PLUG";AT 6,4;"EXHAU  "
 140 FOR G=1 TO 3
 145 PRINT AT 10,16;A$(X)
 150 LET X=X+1
 155 IF X=5 THEN LET X=1
 160 NEXT G
 190 NEXT F
 200 GOTO 100
1000 DIM P$(4,11*32)
1010 DIM L$(11,18)
1020 DIM S$(14)
1030 DIM A$(4)
1032 LET A$(1)="  "
1034 LET A$(2)="  "
1036 LET A$(3)="  "
1038 LET A$(4)="  "
1100 LET L$(1)="                  "
1102 LET L$(2)="                  "
1103 LET L$(3)="                  "
1104 LET L$(4)="                  "
1105 LET L$(5)="                  "
1106 LET L$(6)="                  "
1107 LET L$(7)="                  "
1108 LET L$(8)="                  "
1109 LET L$(9)="                  "
1110 LET L$(10)="                  "
1111 LET L$(11)="                  "
1120 LET P$(1)=S$( TO 7)+L$(1)+S$+L$(2)+S$+L$(3)+S$+L$(4)+S$+L$(5)+S$+L$(6)+S$+L$(7)+S$+L$(8)+S$+L$(9)+S$+L$(10)+S$+L$(11)+S$( TO 7)
1200 LET L$(1)="                  "
1202 LET L$(2)="                  "
1203 LET L$(3)="                  "
1204 LET L$(4)="                  "
1205 LET L$(5)="                  "
```

```
1206 LET L$(6)=""
1207 LET L$(7)=""
1208 LET L$(8)=""
1209 LET L$(9)=""
1210 LET L$(10)=""
1211 LET L$(11)=""
1220 LET P$(2)=S$( TO 7)+L$(1)+S$+L$(2)+S$+L$(3)+S$+L$(4)+S$+L$(5)+S$+L$(6)+S$+L$(7)+S$+L$(8)+S$+L$(9)+S$+L$(10)+S$+L$(11)+S$( TO 7)
1300 LET L$(1)=""
1302 LET L$(2)=""
1303 LET L$(3)=""
1304 LET L$(4)=""
1305 LET L$(5)=""
1306 LET L$(6)=""
1307 LET L$(7)=""
1308 LET L$(8)=""
1309 LET L$(9)=""
1310 LET L$(10)=""
1311 LET L$(11)=""
1320 LET P$(3)=S$( TO 7)+L$(1)+S$+L$(2)+S$+L$(3)+S$+L$(4)+S$+L$(5)+S$+L$(6)+S$+L$(7)+S$+L$(8)+S$+L$(9)+S$+L$(10)+S$+L$(11)+S$( TO 7)
1400 LET L$(1)=""
1402 LET L$(2)=""
1403 LET L$(3)=""
1404 LET L$(4)=""
1405 LET L$(5)=""
1406 LET L$(6)=""
1407 LET L$(7)=""
1408 LET L$(8)=""
1409 LET L$(9)=""
1410 LET L$(10)=""
1411 LET L$(11)=""
1420 LET P$(4)=S$( TO 7)+L$(1)+S$+L$(2)+S$+L$(3)+S$+L$(4)+S$+L$(5)+S$+L$(6)+S$+L$(7)+S$+L$(8)+S$+L$(9)+S$+L$(10)+S$+L$(11)+S$( TO 7)
1500 PRINT AT 3,9;""
1510 PRINT TAB 9;""
1520 PRINT AT 16,9;""
1530 PRINT TAB 10;""
1540 PRINT TAB 5;"SPARK-PLUG "
1550 PRINT AT 21,0;"PRESS ANY KEY TO STOP THE ROTOR."
1600 RETURN
2000 SAVE "ROTAR"
2010 RUN
```

The REM statement at line 0 is entered by a POKE instruction. It is not necessary to key in this line.

Program Description

The 'active' part of the ZX81 computer program, to model the Wankel rotary engine, is really quite short. The animation to show the engine operating, as described above, is created by using four separate rotor and chamber pictures. These are made to follow each other in rapid succession to produce the effect of continuous rotor movement. Subroutine 1000 is used to build up these pictures which are held in the string array P$. Individual lines of each picture are first held in another string array, L$, and each picture is built up from the individual lines with appropriate spacing, S$, added (lines 1120, 1220, 1320 and 1420). This method of building up the large P$ strings from the smaller L$ strings is used to make the entry and editing of the graphics easier to handle.



The four stages of the Wankel's combustion cycle. Top left shows the intake phase followed by the compression 'stroke'. Ignition, above left, is followed by the final uncovering of the exhaust port, allowing spent combustion products to escape and preparing the engine for a new intake of fuel.

The static part of the display is printed out at the end of the subroutine at line 1000. With the picture array established, the loop F (lines 100-190) is used to control the changing display. Within the F loop there is a smaller loop which is used to show the rotation of the shaft using the four graphics characters held in array A$. The program waits at line 110 whenever INKEY$ detects that a key is being depressed to give the user control over the action. The program continues as soon as the key is released. Lines 2000-2010 are used when saving the program so that the program will start automatically whenever it is reloaded.

A cassette version of this program for either ZX81 or Spectrum, together with programs for modelling two and four stroke engines (as published in our November 1982 issue) are available at a cost of £5 from the author at:

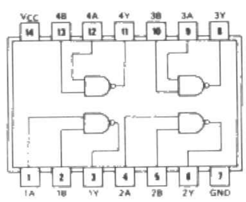
23 Lealholm Crescent
Middlesbrough
TS3 0NA

E&CM

INTEGRATED DATA

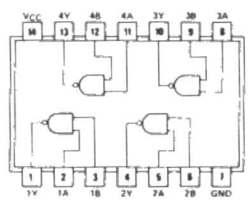
The majority of designs published in E&CM make use of logic devices from the 74xx range of TTL ICs. While our series 'Understanding Digital Electronics' is demonstrating how individual logic gates are interconnected to perform various logic functions, the aim of this new pull-out feature is to detail the logic gates provided by the ICs making up the 74xx series.

1



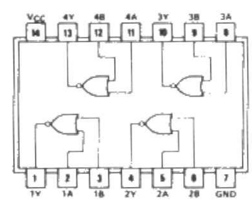
QUADRUPLE 2-INPUT
POSITIVE-NAND GATES

2



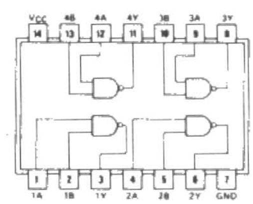
QUADRUPLE 2-INPUT
POSITIVE-NAND GATES
WITH OPEN-COLLECTOR OUTPUTS

3



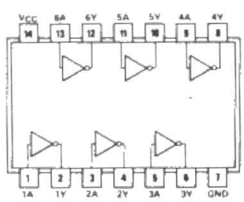
QUADRUPLE 2-INPUT
POSITIVE-NOR GATES

4



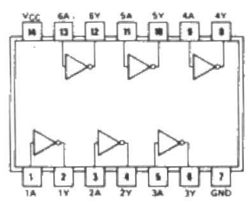
QUADRUPLE 2-INPUT
POSITIVE-NAND GATES
WITH OPEN-COLLECTOR OUTPUTS

5



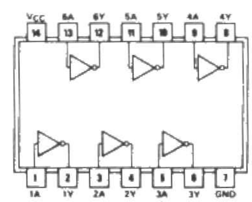
HEX INVERTERS

6



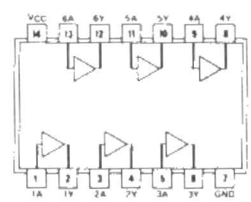
HEX INVERTERS
WITH OPEN-COLLECTOR OUTPUTS

7



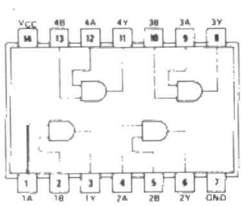
HEX INVERTER BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

8



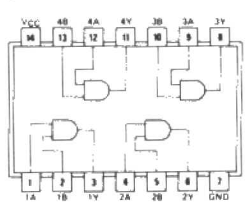
HEX BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

9



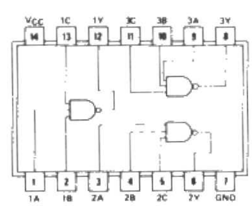
QUADRUPLE 2-INPUT
POSITIVE-AND GATES

10



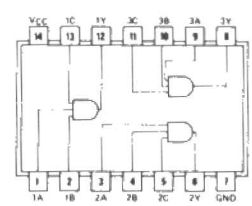
QUADRUPLE 2-INPUT
POSITIVE-AND GATES
WITH OPEN-COLLECTOR OUTPUTS

11



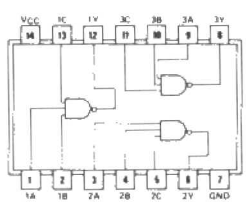
TRIPLE 3-INPUT
POSITIVE-NAND GATES

12



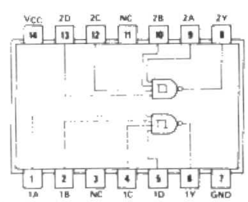
TRIPLE 3-INPUT
POSITIVE-AND GATES

13



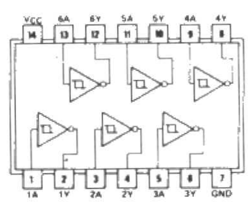
TRIPLE 3-INPUT
POSITIVE-NAND GATES
WITH OPEN-COLLECTOR OUTPUTS

14



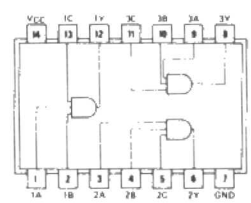
DUAL 4-INPUT
POSITIVE-NAND
SCHMITT TRIGGERS

15



HEX SCHMITT-TRIGGER
INVERTERS

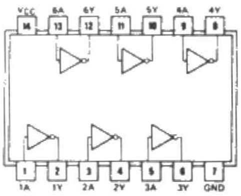
16



TRIPLE 3-INPUT
POSITIVE-AND GATES
WITH OPEN-COLLECTOR OUTPUTS

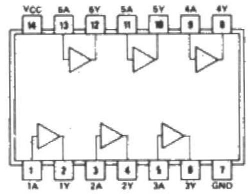
INTEGRATED DATA

17



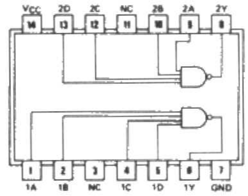
HEX INVERTER BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

18



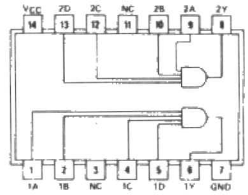
HEX BUFFERS/DRIVERS
WITH OPEN-COLLECTOR
HIGH-VOLTAGE OUTPUTS

19



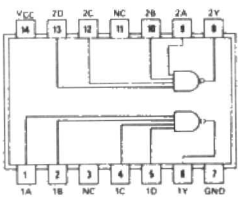
DUAL 4-INPUT
POSITIVE-NAND GATES

20



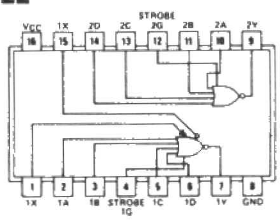
DUAL 4-INPUT
POSITIVE-AND GATES

21



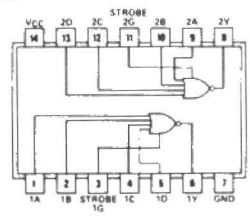
DUAL 4-INPUT
POSITIVE-NAND GATES
WITH OPEN-COLLECTOR OUTPUTS

22



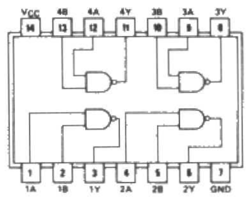
EXPANDABLE DUAL 4-INPUT
POSITIVE-NOR GATES
WITH STROBE

23



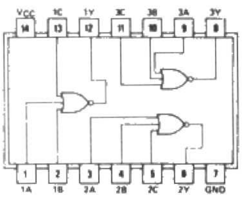
DUAL 4-INPUT
POSITIVE-NOR GATES
WITH STROBE

24



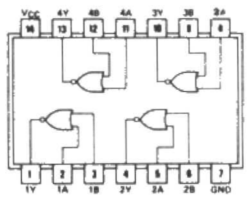
QUADRUPLE 2-INPUT
HIGH-VOLTAGE INTERFACE
POSITIVE-NAND GATES

25



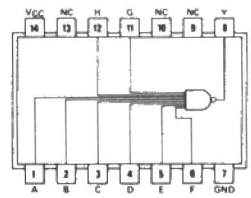
TRIPLE 3-INPUT
POSITIVE-NOR GATES

26



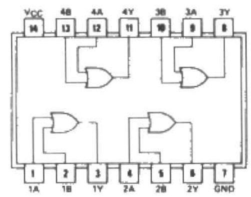
QUADRUPLE 2-INPUT
POSITIVE-NOR BUFFERS

27



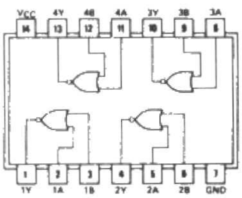
8-INPUT
POSITIVE-NAND GATES

28



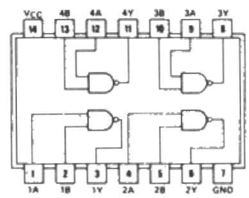
QUADRUPLE 2-INPUT
POSITIVE-OR GATES

29



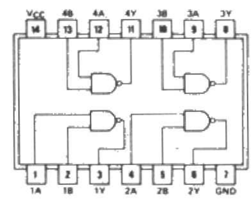
QUADRUPLE 2-INPUT
POSITIVE-NOR BUFFERS
WITH OPEN-COLLECTOR OUTPUTS

30



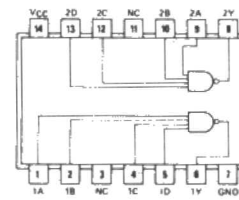
QUADRUPLE 2-INPUT
POSITIVE-NAND BUFFERS

31



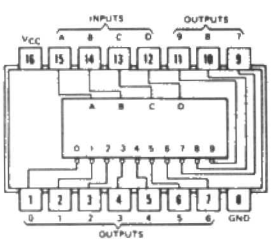
QUADRUPLE 2-INPUT
POSITIVE-NAND BUFFERS
WITH OPEN-COLLECTOR OUTPUTS

32



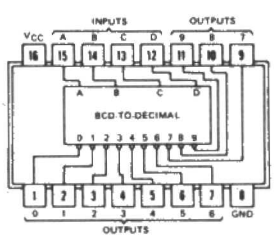
DUAL 4-INPUT
POSITIVE-NAND BUFFERS

33



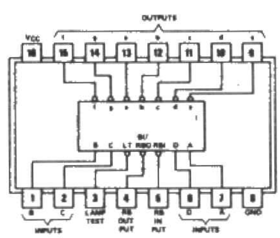
4 LINE-TO-10-LINE DECODERS

34



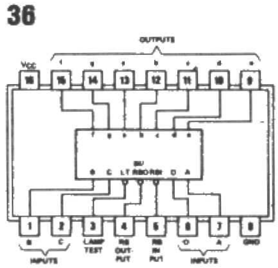
BCD-TO-DECIMAL DECODER/DRIVER

35



BCD-TO-SEVEN-SEGMENT
DECODERS/DRIVERS

36



BCD-TO-SEVEN-SEGMENT
DECODERS/DRIVERS

DRAGON SOFTWARE

S. M. Gee and Kay Ewbank review some Dragon Software that follows on from last month's special feature.

PERSONAL BANKING SYSTEM (£9.95) and **RECONCILIATION MODULE (£5.00)** from HILTON COMPUTER SERVICES LTD., 10 Jerome Road, Larkfield, Kent, ME20 6UR. The first of these programs allows the user to maintain a record of all transactions pertaining to a bank account. When you first use the program you are asked for the date and your current balance. Subsequently you can input new data, correct previous entries, display previous entries, examine standing orders or search through the file for particular entries – by date, cheque number, details or amount. This final facility seems to provide an advantage over the normal paper and pencil methods of keeping track of financial affairs. The program allows you to save and load your own files of data and has a printer routine to enable you to make hard copy.

The BANK RECONCILIATION MODULE is intended to be used in conjunction with this program to compare the statements you receive from the bank with the records kept on your Dragon. The instructions offer software maintenance, that is any additional enhancements to the program, to purchasers of the program who register their names and addresses. All in all this is a professionally finished, user-friendly piece of software.

DEMON (£18.95) (cartridge) and **SPACE RACE (£7.95)** from COMPUSENSE, 286D Green Lanes, Palmers Green, London N13 5TN. We were very impressed by the machine code monitor contained on the DEMON cartridge. It provides a facility for writing your own machine code programs and makes the transition to programming in assembler really straightforward. The instructions are easy to understand even for a beginner and several example programs are provided and documented. Compusense also market an assembler. Both these cartridges will appeal to the serious Dragon programmer who wants to go beyond BASIC.

We had lots of difficulty loading the cassette tapes from Compusense and were glad to see that SPACE RACE was also available in cartridge form which makes loading virtually instantaneous and entirely trouble-free. The object of this moving graphics game is to destroy three types of aliens by firing missiles from your space ship. This is a machine code game which can be played either from the keyboard or with joysticks but we found it to be more responsive to the keyboard. There are fifteen difficulty levels so you can suit the game to your skill – the aliens move faster according to the factor you select. This game makes effective use of the Dragon's sound capabilities and was fun to play.

GOLF (£7.95) from AUDIOGENIC, 34-36 Crown Street, Reading. This game of golf is for one or two players and the players can enter their handicaps at the beginning of the game. There are two decisions to make about each pitching stroke – the choice of which wood or irons (numbered correctly in this case) and the angle of the shot. Once you are on the green you have to decide the angle and distance of your shot. It takes a long time for each hole to be displayed and the graphics display seems to be unnecessarily complex. On the whole we still prefer to play golf out-of-doors!

LIGHT PEN (£10.00) from TROJAN 166, Derlwyn, Dunvant, Swansea, SA2 7PF. This package combines hardware – the pen itself – and software to demonstrate its potential. The demo program has four options, a game of Nim – which is indeed more fun to play with the light pen than the alternative wordy versions – a light graph – which shows via a bar chart the intensity of light given off by the three primary colours and the green background, a sound and light option – which produces notes whose pitch varies corresponding to the light intensity of all eight of the Dragon's available colours – and the instructions for the use of the light pen with examples of its use in multiple choice-type tests and details of how to incorporate light pen routines into your own programs. The light pen itself is fairly responsive but can be made to respond to spurious signals in the environment! It would probably be a useful tool in any educational setting.

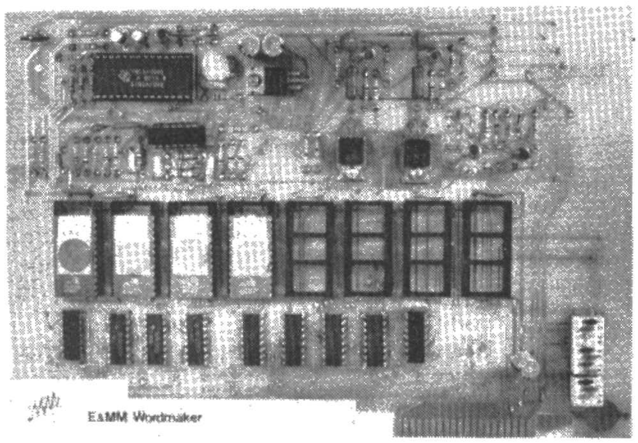
DRAGON TREK (£9.95) from SALAMANDER SOFTWARE, 17 Norfolk Road, Brighton, BN1 3AA. This is a nicely produced implementation of the game based on the popular television series Star Trek that comes with extensive documentation. It has attractive graphics and makes good use of sound effects. If you are a devotee of Captain Kirk, Scotty, Mr. Spock and Lt. Uhura and want to join in their fight against the Klingons then this is a program well-worth considering.

RING OF DARKNESS (£10.00) from WINTERSOFT, Westminster Bridge Road, London SE1. This is an intriguing adventure game with an attractive graphics display. It consists of a number of program modules and the initial instructions, which guide you through the setting up are clear and user-friendly. At the beginning of the game you have to choose how to distribute forty points between intelligence, strength and agility. Next you have to choose the character type you wish to assume – you can be an elf, a dwarf or a human – and your character skill – be it warrior, thief or wizard. Having made these decisions you load the game, which takes quite a long time, and a map of the enchanted land appears. This tape should appeal to adventure game enthusiasts.

*** All prices are inclusive of VAT, postage and packing.**

SPEECH AND MUSIC

The ability to generate 'sound' of one sort or another is an important feature of many micro systems. In part 5 of our series 'Understanding Your Computer', B. Boyde-Shaw explains how to 'make music' on some popular machines.



E&MM Wordmaker

Sound can, in essence, be divided into one of three categories: speech, music or noise, and what is noise to one person, can equally well be music to the ears of another. It all depends on the frequency used, the number of frequencies used at any one time, the length of time the frequency is sounded and finally, the volume.

In a computer it is usual to be able to program these four qualities (parameters) when setting up a sound sequence. **Table 1** shows typical 'sound' commands for three popular micros. The command parameters control, in the case of D the Duration of the note, A the Amplitude or noise level, P the Pitch or frequency, and C the Channel number where more than one note is available (chord production possible). Many computers today provide 3 pure notes and one noise channel that can all be sounded simultaneously.

TABLE 1

SOUND	C, A, P, D
CALL SOUND	(D, P, A)
BEEP	(D, P)

Values for the amplitude vary from computer to computer and do not follow a standard. The volume provided is also dependant on the machine's internal loudspeaker or the volume setting of a television monitor. The Texas for example does not have an internal speaker, the BBC and the Spectrum do, but the Spectrum's sound channel can be outputted through the MIC or EAR socket for further amplification.

Some computers are more realistic than others in the code chosen for the pitch or frequency number. Texas use 262 as middle C, whereas the BBC uses 101 and the Spectrum uses 0, and thus pitch parameters range positively and negatively either side of 0 to achieve other notes.

Many micros provide a facility to prematurely cut off a note so as to produce the very useful stacatto rhythm or sound. On the more sophisticated computer, notes can be encased in an 'envelope' so as to produce a more realistic sound, by, for example, varying the pitch while the notes are being played. This is achieved by dividing the duration of a sequence into steps – changing the pitch from one value to another while that tuned section is running. For example, a wailing sound can be produced by programming an ENVELOPE command, to change the pitch of a note in given steps at a given time, but differently for each timed section as in **Fig 1.**

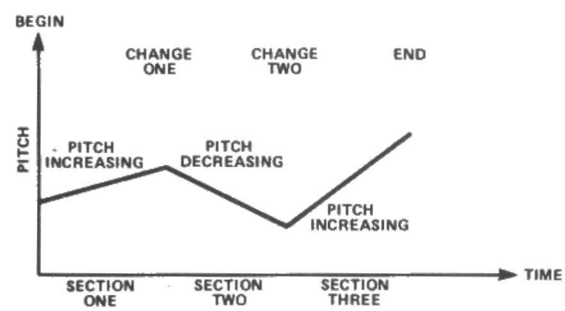
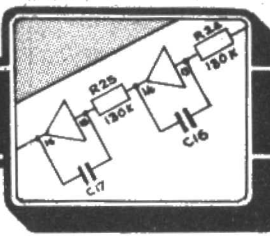


Figure 1. An ENVELOPE command can be used to change the pitch of a note.

The statement 'envelope' can also be programmed to change amplitude while a note is being played, in the same way as the pitch was varied. For example in **Fig 2** we could represent a plane flying overhead by first choosing a suitable note for the sound of the plane, then increasing its volume as it came towards us and decreasing it as it went away.

Naturally because of the multiple channel functions on some computers (not the Spectrum) we can play chords and on a sophisticated machine (e.g. the BBC), synchronisation of notes on various channels is also possible.

The Texas computer allows the programmer to achieve a chord in one 'call sound' statement, whereas the BBC uses a synchronisation sequence and the notes for the chord are programmed on successive lines.



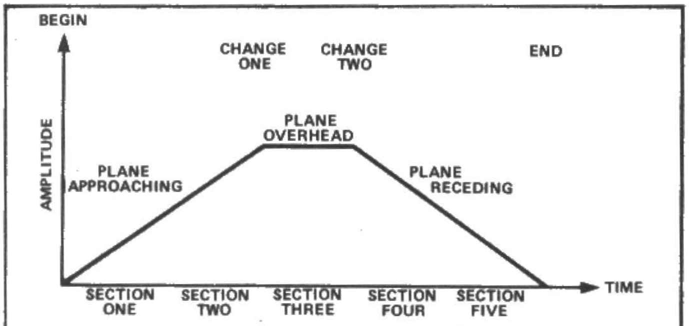


Figure 2. The sound profile of an overflying aircraft.

Fred the dog, in the last article, could be made to bark and scratch his claws on the pavement by experimenting with the notes on your computer (including any pre-programmed noises) and putting the sound lines in the correct position in the program. Once you have worked out how to make him run towards you and then to run away after the 'Back Fred' command you could 'envelope' your sound commands to produce the correct approaching and receding sounds.

Making A Speech

As yet you need a speech synthesizer to get your computer to talk to you while your program is running. They often come in the form of a ROM package, which is either fitted inside the computer or into an external socket. You are usually limited to programming in the words that the computer manufacturer has decided would be the most useful. Most packages will provide all the numbers needed to make up single and multiple numbers, i.e. 20+1 gives you *'twenty one'*: 100 + and + 20 + 4 gives you *'one hundred and twenty four'*. In addition the natural colours, the alphabet, conjunctions, simple verbs and pronouns will be available. The rest will be determined by the manufacturer who considers what you may want to be doing with the device before deciding on the other words to give you, which may be around 500 in all.

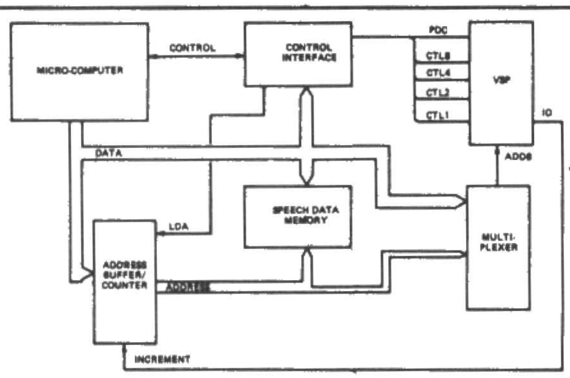


Figure 3. Block diagram of the ROM based speech synthesizer shown at the start of the article.

Another method is to break the words down into their component parts and give each part a code. By this method you can build up your own words and increase the machine's vocabulary tremendously, and also have one more fitting to your requirements.

For example, take just a few sounds such as:

ne, ou, er, tai, ss, ke, wai, te.

From these sounds we could make up the sentence:

'Wait to take ten curtains, Kent'

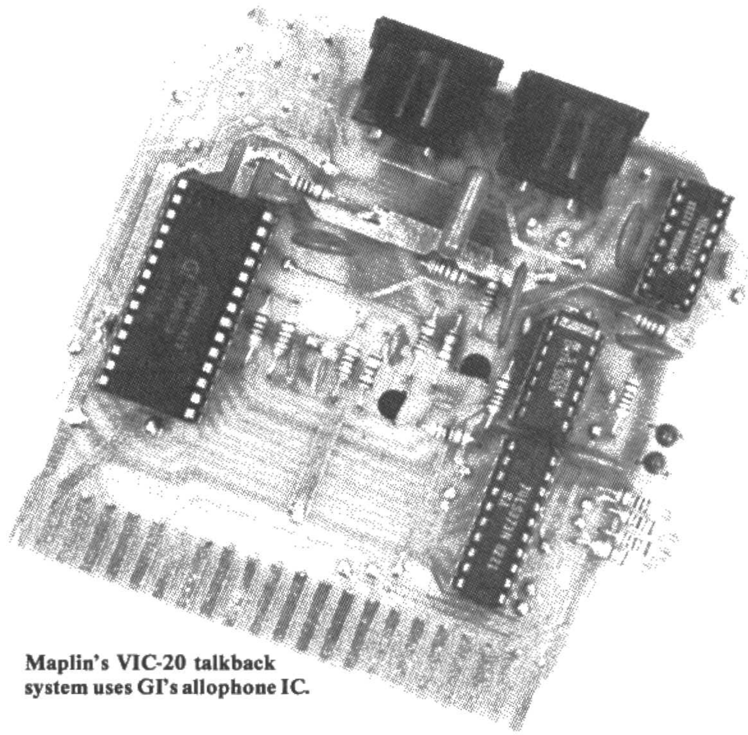
wai,te, ,te,ou, ,tai,ke, ,te,ne, ,ke,er,te,ne,ss, ,ke,ne,te

Notice the programmed spaces.

It may seem strange to use 8 sounds to produce 6 words, these same sounds can however make up other words and phrases without any more ROM coding, for example:

'Ken wastes ten, Kurt earns ten too'.

Work this sentence out now for yourselves, using the sounds above.



Maplin's VIC-20 talkback system uses GI's allophone IC.

Obviously the speech would be a little stilted, and over pronounced, but quite understandable. In this way the computer could give out instructions vocally instead of writing them on the screen – ideal if you can't yet read, for example, playgroups and reception classes in infant schools.

Getting back to our games with Fred, at the moment he is running here and there, barking and scratching the pavement, but his unseen master only writes on the screen 'Back Fred'. Now of course with a speech synthesizer in our computer we could make the master speak the words 'BACK FRED', providing we had those words in the computer's vocabulary or we were using the later method using Be, aa, ke, fe, red. Using speech in games can make the game much more effective. The shock effect of adverse situations can be increased in the same way that explosions, Zaps and loud laser firings can (these sounds, incidentally, are pre-programmed on the ORIC 1).

Given a year or two we will be able to buy computers with a built in speech function and probably (or is it hopefully) at less than £100. The next step is to get the computer to understand what you are saying and so do away with that one device that prevents the computer getting any smaller, that's right, the keyboard.

Next month – more on designing games.

E&CM

EXPERT SYSTEMS

In this month's Computer Brain, Mike James looks at programs that think. These programs are beginning to find practical application in the form of so-called Expert Systems.

If you have been following this series you might have come to the conclusion that Artificial Intelligence seems to be about writing programs that replicate a very small part of human behaviour. For example, at the start of the series we looked at how computers can play games, we then looked at how computers can deal with language and in the most recent articles we examined that difficult area of computer vision. However surely the main problem of A.I. is understanding and writing programs that 'think'. To a certain extent programs that play games such as chess are attempts at carrying out tasks that humans would say need thought. The trouble is that once you know how such programs work it is difficult to believe that they play in the same way that humans do. Do grand masters really analyse chess programs six to ten moves ahead? Aren't chess programs simply using the computer's speed to do a 'brute force' analysis of the game? In the same way that it is clear that chess programs are not playing chess in the same way that humans do so, many A.I. programs seem to achieve roughly the same result as a human but by methods that seem very different. It is important to realise that whenever you come to understand the way that an A.I. program works you are bound to think that its operation is essentially simple – anything you understand seems simple. All this seems to lead to the conclusion that perhaps the A.I. programs that we have looked at so far are simply clever programs rather than intelligent programs! This is in part true but it does ignore some important points. The human brain is a very complicated device and it is quite capable of carrying out a large number of very simple operations at the same time and it is quite possible that human intelligence is the result of many simple operations carried out simultaneously or repeatedly. What this means for our study of artificial intelligence is that we may already know how a human does something but have no way of actually using this knowledge to write a program because no digital computer could run it in a reasonable time. In other words the choice may be to do it as a human does it and not have a practical program or take advantage of the particular talent of the digital computer and do it in a non-human way. So in the case of chess playing it is likely that human players do look a number of moves ahead, but they do not examine all the possible moves. Instead they use a heuristic to 'prune' the move tree down to a manageable size. Most computer chess programs examine all the moves in the move tree to find a good move and this is the way a computer will very often come up with a move that is, to a human, not part of any playing 'strategy'.

So do computer chess programs in any sense 'think' about the game they are playing? To a certain extent the answer to this question depends on how cautiously you use the word 'think'. The chess program is probably using some of the methods that a human uses – weighing up moves, examining some parts of the move tree etc. But to make the chess program practical it uses the unique advantages of the digital computer to do arithmetic and search the move tree fully to a given depth. In my opinion chess programs do not think about chess but they do share things in common with a human player.

It is the same story with other A.I. programs, they use some human methods but combine them with methods that suit a computer. Now it may be that one day programs will be written that use all human methods but I feel that this is unlikely in the near future. For the time being we will write programs that play chess or solve problems using a certain amount of cleverness and mostly brute force.

This is not to say that we cannot write programs that give the appearance of thinking. In the same way that we write chess programs that play chess to grand master level it is quite possible to write programs that 'think' and solve problems in specific areas as well as experts.

A General Problem Solver

It is not difficult to write a program that solves problems of a specific type. For example you could write a program that would prove mathematical theorems. Indeed researchers in A.I. have written programs that 'reason' about areas of mathematics that are so good that they occasionally find a better proof of a known theorem or even find and improve a theorem that was previously unknown. In this sense we can say that A.I. programs have already surpassed man's intelligence! Of course if you take a maths theorem proving program and ask it something unconnected with mathematics you won't get a reasonable response! The program works within a very small 'world' and it cannot move outside to consider anything else. However in the early days of A.I. research it was shown that a general purpose logical reasoning program could be written. All you had to do was give it a description of the 'world' that it had to reason about and off it would go, solving logical problems with no trouble.

Humans Solving Problems

Computers reason using exact logic and this is why they are so good at proving mathematical theorems. However humans rarely think logically in this sense unless they are forced to by the nature of the problem. This is not to say that humans think illogically! What tends to happen is that a human will not analyse the situation in minute detail but try to draw on past experience. He or she will try to match some detail of the problem with something that they have

seen in the past. In other words they use a wide ranging knowledge of the way that the world works, knowledge that is generally included in programs that reason logically.

If this is that case then to construct programs that reason about things in the way that humans do we should first look at the representation of knowledge inside a computer. Computers are thought to be good at collecting and storing vast amounts of information and indeed this is true, but they store it in a very simple way. A computer's collection of facts is more like the way that an encyclopedia 'remembers' things rather than the way a human remembers. For example a collection of facts is useless unless you know what the 'consequences' of the fact are. If you are trying to decide what the weather is both you and the computer might know that the sky is black and full of clouds but you can deduce that this means that is likely to rain. In other words you know the (possible) consequences of a black cloud filled sky but the computer does not!

It is not difficult to think of ways of storing information along with its consequences as a collection of rules. For example the weather 'knowledge' could be stored inside a computer as –

IF black cloudy sky THEN high possibility of rain

In general a piece of knowledge can be represented by a list of conditions and a list of consequences. For example –

IF black clouds, high humidity, summer THEN thunder storm

might be a statement of what you know about thunder storms! Notice that although we are using IF . . . THEN which is so familiar from programming, this use is different. The IF . . . THEN in this case isn't an instruction to do something if something is true, it is a statement of the relationship between different facts.

If you wanted to construct a program that would use such rules to solve problems then all you would do is to collect as many rules as possible, in other words build a 'rule data base', and then, to find out the meaning or consequence of a set of conditions, simply search the data base for rules with the same conditions. There may be more than one rule for any particular set of conditions. For example, if you knew that there was a black sky you might search and find both of the IF . . . THEN weather rules given above. The predicted consequences would be rain by the first rule and possibly a thunder storm from the second. To find out which you would have to supply more information.

Programs of this sort are usually referred to as 'knowledge-based expert systems' and they are receiving a great deal of attention from the computer community and the general public at the moment as the best thing that A.I. has ever produced. Now you know that such programs are not at all complicated! The main problem with any expert system is the collection of the rules that form the data base. This is usually accomplished by working with a human expert and trying to find out what rules he uses.

Rather than continue with theory and explanation it is easier and more fun to go straight to a simple BASIC expert system.

The Aardvark Program

All this talk of knowledge rules and expert systems may seem convincing, but does it work? To demonstrate how powerful the idea is, the program given below will learn to become an expert on types of animals. The reason why types of animal has been used is that this particular program has a long history in one form or another and has always been presented with an animal data base. However, since the program learns the data base rather than having it already built in you could use it in other areas such as fault diagnosis just by changing the first question asked. The reason that this is a very simple expert system is that it uses rules of the sort

IF list of animal characteristics THEN it is a 'animal name'

For example you might have the rule

IF it has feathers, is a bird of prey THEN it is an eagle

To make finding the correct rule easy and to make the addition and modification of existing rules possible all of the rules are represented by our old friend the tree (see last October's *E&CM.* Vol. 2, Issue 10).

To see how a tree can be used to represent a number of rules look at **fig. 1.** Starting from the first question "Does it have feathers" you can work your way down the tree answering the questions until you come to an animals name. Each time you answer a question you take the branch of the tree that corresponds to the answer i.e. the lefthand branch for yes and righthand branch for no and ask the next question you meet. For example if the answer to "Does it have feathers" is no then the next question is "does it have wool". If the answer to this question is yes then the animal is a sheep. The rules that are contained in this tree are –

IF feathers, bird of prey THEN eagle
IF feathers, not bird of prey, brown THEN sparrow
IF not feathers, wool THEN sheep

You should be able to see that each of these rules is represented in the tree by the path that you have to take to get to the animals name. The advantage of storing the rules in this form is that you can match the conditions against the rules one at a time rather than all together. It also provides a way of asking the user to supply information when it is necessary rather than all at once.

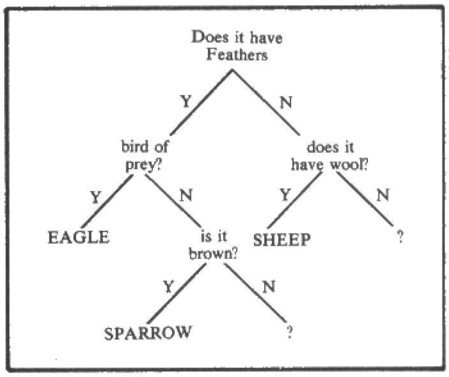


Figure 1. Decision tree mentioned in text.

This is all very well but how do the rules and the tree structure get there in the first place? The answer is that every time the Aardvark program reaches a "?" in the tree structure it doesn't know what the animal is. To put this to rights and gain some information it asks the user what the animal is called and for a question that it can ask next time to identify the animal. Once it has this information it inserts it into the tree structure for later use. For example, suppose after asking a number of questions the program finds itself at the "?" following the "is it brown" question. It then asks the user "what is the animal" to be told that it is seagull and the question it should ask is "Is it a sea bird?". The result is that the new question replaces the question mark and

the yes branch of the new part of the tree leads to "seagull" and the no branch to yet another question mark.

There is one other way that the Aardvark program can learn and that is by getting the answer wrong! If it follows a path down the tree and arrives at the answer "sparrow" only to be informed by the user that the animal is in fact a "wren" then it can avoid this mistake a second time by asking the user for another question to ask to tell the difference between the two birds. For example if the question is "is it the smallest brown British bird" then the "wren" would be on the yes branch and the "sparrow" on the no branch. The question itself would of course replace the entry "sparrow" in the original tree. In this way the tree 'grows' and modifies itself to reflect what you tell it about animals. You will have to try it for yourselves to see just how quickly it learns but it is great fun to use!

The program is written in a standard Microsoft BASIC and should run on almost any machine without modification apart from the ZX81 and the Spectrum, which will require some slight changes to the string handling.

The details of the program are not difficult to understand. The subroutine structure is –

10-70	Initialisation
80-120	Main program loop – once round per animal
1000	Initialisation subroutine
2000	Ask question and process answer
3000	Answer input routine
4000	Don't know animal so get new animal and question
5000	Report animal found and check correct
6000	Incorrect guess so get new animal and question

The only other information that it is important to know is that the tree structure is represented by two arrays R and L corresponding to the left and right branch of the tree following each question. If you reach an element of either L or R that contains zero then you have reached the end of the tree and don't know what the animal is. If you reach an element of either L or R that contains a negative number you have found a possible candidate for the animal. The names of the animals are stored in the array N$ and the index of the animal that you have found is stored

Sample of program output

(for clarity user's responses are given in lower case)

```
AARDVARK
YOU THINK OF AN ANIMAL AND
I WILL GUESS IT -
ANSWER EACH QUESTION WITH
YES OR NO

DOES IT HAVE FEATHERS
? no
I DO NOT KNOW THE ANIMAL
THAT YOU ARE THINKING OF
WHAT IS IT CALLED
? dog

WHAT EXTRA QUESTION CAN I ASK
TO DISTINGUISH THIS NEW ANIMAL
? does it bark
FOR A DOG
WHAT IS THE ANSWER TO
DOES IT BARK
? yes

THINK OF A NEW ANIMAL

DOES IT HAVE FEATHERS
? no
DOES IT BARK
? no
I DO NOT KNOW THE ANIMAL
THAT YOU ARE THINKING OF
WHAT IS IT CALLED
? cow

WHAT EXTRA QUESTION CAN I ASK
TO DISTINGUISH THIS NEW ANIMAL
? does it eat grass
FOR A COW
WHAT IS THE ANSWER TO
DOES IT EAT GRASS
? yes

THINK OF A NEW ANIMAL

DOES IT HAVE FEATHERS
?no
DOES IT BARK
? no
DOES IT EAT GRASS
? yes
IS IT A COW ? n
I GIVE UP!
WHAT IS IT ? sheep
WHAT QUESTION WOULD TELL
THE DIFFERENCE BETWEEN A COW
```

```
10 PRINT "AARDVARK"
20 PRINT "YOU THINK OF AN ANIMAL AND"
30 PRINT "I WILL GUESS IT -"
40 PRINT "ANSWER EACH QUESTION WITH"
50 PRINT "YES OR NO"
60 PRINT
70 GOSUB 1000

80 GOSUB 2000
90 PRINT
100 PRINT "THINK OF A NEW ANIMAL - "
110 PRINT
120 GOTO 80

1000 DIM Q$(20),R(20),L(20),N$(20)
1010 Q$(1)="DOES IT HAVE FEATHERS"
1020 D=1
1030 R(1)=0
1040 L(1)=0
1050 N=0
1060 Q=1
1070 RETURN

2000 X=1
2010 GOSUB 3000
2020 IF A$="Y" AND R(X)=0 THEN GOTO 4000
2030 IF A$="N" AND L(X)=0 THEN GOTO 4000
2040 IF A$="Y" AND R(X)<0 THEN A=-R(X):GOTO 5000
2050 IF A$="N" AND L(X)<0 THEN A=-L(X):GOTO 5000
2060 IF A$="Y" THEN X=R(X)
2070 IF A$="N" THEN X=L(X)
2080 GOTO 2010

3000 PRINT Q$(X)
3010 INPUT A$
```

```
3020 A$=LEFT$(A$,1)
3030 IF A$="Y" OR A$="N" THEN RETURN
3040 PRINT "I DO NOT UNDERSTAND YOUR ANSWER"
3050 PRINT "PLEASE ANSWER YES OR NO TO MY"
3060 PRINT "QUESTIONS - THANK YOU"
3070 PRINT
3080 GOTO 3000

4000 PRINT "I DO NOT KNOW THE ANIMAL"
4010 PRINT "THAT YOU ARE THINKING OF"
4020 PRINT "WHAT IS IT CALLED "
4030 INPUT B$
4040 PRINT
4050 PRINT "WHAT EXTRA QUESTION CAN I ASK"
4060 PRINT "TO DISTINGUSH THIS NEW ANIMAL"
4070 INPUT C$
4080 N=N+1
4090 N$(N)=B$
4100 Q=Q+1
4110 Q$(Q)=C$
4120 IF A$="Y" THEN R(X)=D+1 ELSE L(X)=D+1
4130 PRINT "FOR A ";B$
4140 PRINT "WHAT IS THE ANSWER TO"
4150 PRINT C$
4160 INPUT D$
4170 D$=LEFT$(D$,1)
4180 IF D$="Y" OR D$="N" THEN GOTO 4300
4190 PRINT "ANSWER YES OR NO PLEASE"
4200 GOTO 4130
4300 D=D+1
4310 IF D$="Y" THEN R(D)=-N:L(D)=0
4320 IF D$="N" THEN L(D)=-N:R(D)=0
4330 RETURN

5000 PRINT "IS IT A ";N$(A);
5010 INPUT B$
5020 IF LEFT$(B$,1)="N" THEN GOTO 6000
5030 IF LEFT$(B$,1)<>"Y" THEN PRINT "ANSWER YES OR NO PLEASE":GOTO 5000
5040 PRINT "I THOUGHT SO !!"
5050 PRINT
5060 RETURN

6000 PRINT "I GIVE UP !"
6010 INPUT "WHAT IS IT",B$
6020 PRINT "WHAT QUESTION WOULD TELL"
6030 PRINT "THE DIFFERENCE BETWEEN A ";N$(A)
6040 PRINT "AND YOUR ";B$
6050 INPUT C$
6060 Q=Q+1
6070 Q$(Q)=C$
6080 N=N+1
6090 N$(N)=B$
6100 IF A$="Y" THEN R(X)=Q ELSE L(X)=Q
6110 D=D+1
6120 PRINT "FOR A ";B$
6130 PRINT "WHAT IS THE ANSWER TO"
6140 PRINT C$
6150 INPUT D$
6160 D$=LEFT$(D$,1)
6170 IF D$="Y" THEN R(D)=-N:L(D)=-A
6180 IF D$="N" THEN L(D)=-N:R(D)=-A
6190 PRINT
6200 RETURN
```

in the L or R as a negative number. So if you find that L(X) is negative for instance the animals name is N$(-L(X)). If however the value stored in R or L is positive and not zero then this is the index of the next question you should ask and the index of the next element of either R or L you should look at as a result of the answer to the question. Finally the questions are stored in the array Q$.

The tree starts off with one question and no animal names stored. So Q$(1) contains the question and R(1) and L(1) both contain zero (you can think of zero as standing for the question mark in the tree diagram in **fig. 1**). If the answer to the first question is yes then R(1) is examined and as it is zero the program asks for an animal name and a new question. The new name is stored in N$(1) and the question in Q$(2). R(1) is changed to 2 so that question Q$(2) will be asked following a yes answer to question 1 and R(2) and L(2) are changed so that one of them holds −1 to indicate that the animals name is in N$(1) and the other is set to zero to indicate that yet another question and name is needed.

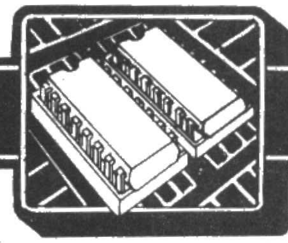
More Work On Aardvark

You might like to change the first question in Aardvark to something else like – "is the fault electrical?" to turn the program into a fault finder. Of course naming the fault is only part of the solution. You also should store instructions about what to do along with the name. Aardvark is not as sophisticated as a fully commercial expert system – it doesn't give the probability of something being true for example – but it is fun and it can form the basis for your experiments. Some suggestions for facilities that you might like to add to Aardvark are –

- add a routine to save and read in an existing tree
- find a way of printing the entire tree for analysis
- allow the user to ask for a description of any animal in the tree

a more advanced project would be to write a program that would take a 'knowledge tree' as produced by Aardvark and process it so that the minimum number of questions are asked to locate a particular animal. The next most useful thing to add to Aardvark would probably be probability! But for this you will have to wait for the next episode of "The Computer Brain".

Next month – fuzzy thinking!



SPECTRUM MONITOR

John Williams describes the addition of a 4K EPROM to either a 48K or 16K Spectrum. The technique described avoids any conflict between RAM and the additional memory.

The standard Spectrum provides all the facilities needed to write and run BASIC programmes, however, difficulties arise when one wishes to speed up parts of the program by using machine code. This can be accomplished on the standard machine but to get the code into memory requires some form of monitor or hexadecimal loader. It is not always convenient to load the monitor from tape because this takes time, occupies memory and may conflict with the line numbers of the program being written. The circuit to be described provides 4K bytes of EPROM which can hold a resident monitor for instant use and still leave plenty of room for some of the programmer's commonly used subroutines.

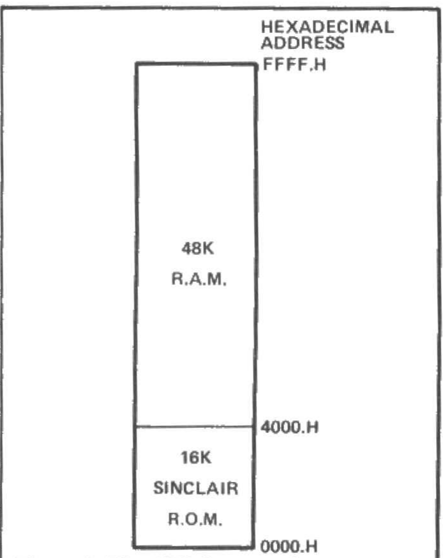


Figure 1. The 48K Spectrum's memory map.

Principles

Whilst it is relatively straightforward to add memory to a ZX81 or 16K Spectrum a look at the memory map of the 48K Spectrum in **Fig 1** reveals that there are no unused addresses available for further memory. It is however possible to share a part of the memory map between the EPROM and the Sinclair ROM as shown in **Fig 2.** This can be done only because there is a 1K block in the ROM between 3C00 and 3FFF which contains no Z80A program instructions, but instead a lookup table. This contains the character set that defines the graphic and alphanumeric shapes that are displayed on the screen. The display program itself is elsewhere in ROM, and in consequence, program instructions will never be read from this 1K block in normal operation. If such a read occurs, the circuit detects that EPROM is required instead of ROM. A bistable is set and all further reads will be from EPROM until such time as a program instruction is read from outside the EPROM range. These program instruction reads are distinguished from other reads by the M1 line going to its active low state, the address bus identifies the memory area in use. By this means address sharing is achieved in a way that is transparent to the user.

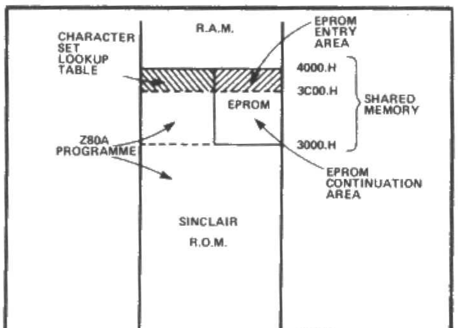


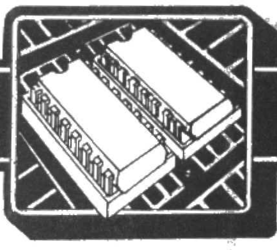
Figure 2. It is possible to share part of the Spectrum's memory map.

It will be seen that the EPROM can only be entered in the part 3C00 to 3FFF but once this is done programs are free to use all of the 4K down to 3000. It is however necessary to stop the microprocessor interrupts by a disable interrupt DI instruction before doing this. Interrupts occur every 20ms and when enabled cause a ROM subroutine to be run at 0038. This ends with a return instruction when control is passed back to the main program at the point where the interrupt occurred; if this was below 3C00 the EPROM will not be reselected. Therefore the DI instruction must be used before going below 3C00. Only the part of the program above 3C00 can call subroutines outside the EPROM and it is advisable to enable the interrupts before calling ROM subroutines as some of them require interrupts to operate. Any part of the EPROM can read or write to RAM. To avoid filling the easier to use area, the monitor program has been kept below this except for calls to the ROM. There are 960 bytes available for the users programs, and a further 2 1/2 K below 3970.

Detailed instructions on how to program the EPROM are not given here, however the design of a programmer to use with a Spectrum was published in the February 1983 issue of *Electronics & Computing.*

Construction

The construction of the unit is not critical and a variety of techniques will prove satisfactory. A piece of Veroboard 3¾ inches square is ample for this project. A socket must be used for the EPROM as this may be required to be removed for erasure and reprogramming from time to time. The use of sockets for the other ICs is a valuable aid to fault-finding. The board is connected to the computer via a 28 way double sided edge connector, the pin connections of which are given on page 180 of the Spectrum manual. The connector may be mounted on the board or be connected by not more than one foot of ribbon cable. The circuit draws up to 170 mA depending on the make of EPROM and whilst this can be drawn from the Spectrum 5 volt rail a separate regulator is advisable for continuous use, and is included in the design. This draws power direct from the power unit via the edge connector and does not cause any extra heat to be generated within the computer.



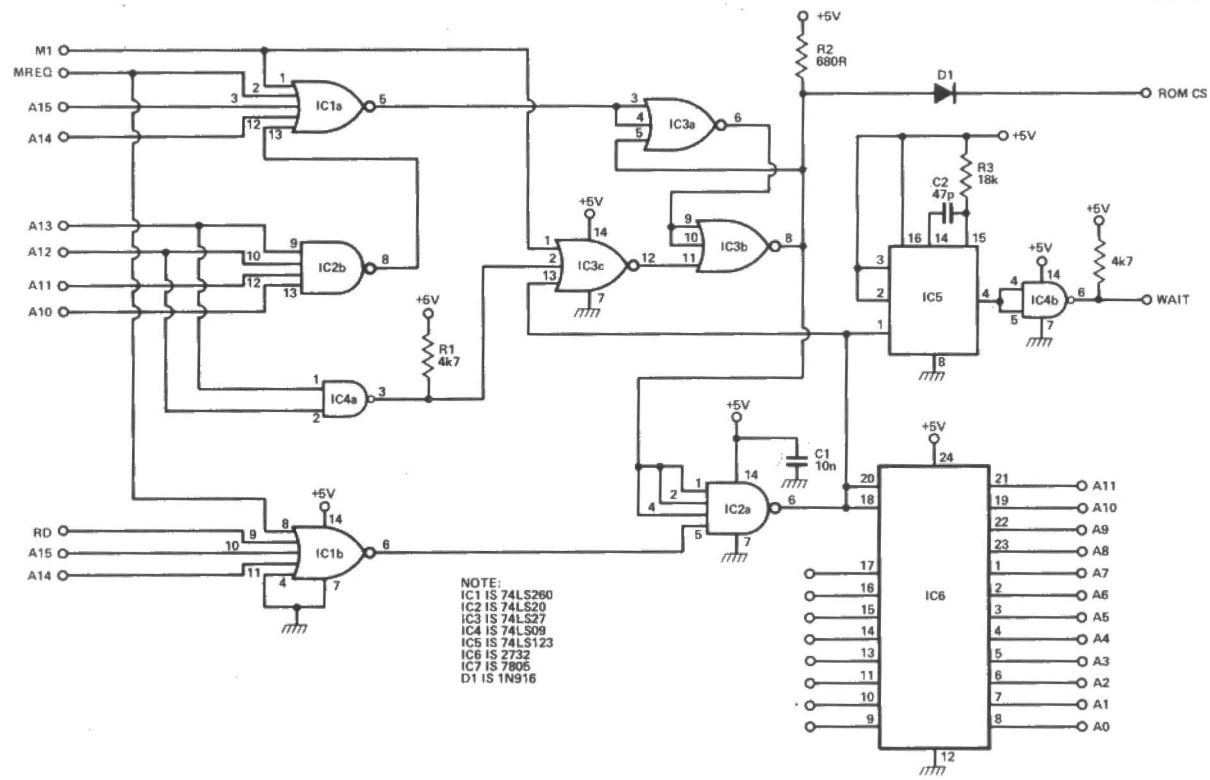


Figure 3. The Spectrum monitor's circuit diagram.

PARTS LIST

Resistors	
R1, 4	4k7
R2	680R
R3	18k
Capacitors	
C1	10n
C2	47P
C3	470n
C4	220n
Semiconductors	
IC1	74LS260
IC2	74LS20
IC3	74LS27
IC4	74LS09
IC5	74LS123
IC6	2732
IC7	7805
D1	1N916
Miscellaneous	
DIL IC sockets, edge connector, TO220 heatsink, etc.	

Testing

Once built, a programmed EPROM will be required in order to check the board. To aid in the location of any possible faults it is recommended that a very simple program is put into the EPROM and that the hardware is checked using this. When the board is proved to be operating satisfactorily the full software can be used and any subsequent problems be assumed to be due to errors in typing in the program.

A test program is given in **Fig 4,** this is to be programmed into the EPROM starting at hex address 3C30 which is decimal 15408. A loader program to drive the EPROM programmer referred to above is shown in **Fig 5.** After checking the board wiring, temporarily disconnect one end of D1 and plug the board to the Spectrum but leave out the ICs. Switch on and check that the copyright message is displayed as usual. A short on the input wiring is the most likely cause of failure at this stage. Plug in ICs 1, 2, 3 & 5, switch on and again check for the message. If this does not occur remove the ICs one at a time to identify the culprit and check the wiring again in that area. finally insert the remaining ICs and reconnect D1. Any further faults will require careful wiring checks. As a guide if test equipment is available, when operating in BASIC IC3 pin 8 will be logic low, IC2 pin 6 and IC5 pin 4 will be logic high.

Once it has been verified that the computer is performing normally in BASIC the EPROM can be tried out. PRINT USR 15408 should result in the number 99 being printed to the screen. This calls the test program which simply loads the BC register pair with 99 and returns to BASIC which prints the value. This program is identical to the example in chapter 26 of the Spectrum manual. Although this program does not exercise all the functions of the hardware there is nevertheless a good probability that if it operates correctly so will the full software package.

```
3C30   01 63 00 C9
```

Figure 4. A test program for the monitor board – short and sweet.

```
10 LET a=15408
15 RESTORE
20 READ c$
30 LET c=CODE c$-48-(7*(c$(1)>"@"))
40 LET d=CODE c$(2)-48-(7*(c$(2)>"@"))
50 POKE a,16*c+d
60 LET a=a+1
70 IF LEN c$=2 THEN GO TO 20
80 LET c$=c$(3 TO )
90 IF c$(1)<>" " THEN GO TO 30
100 LET c$=c$(2 TO ): GO TO 90
110 REM Must use CAPITALS in Data lines
200 DATA "01 63 00 C9"
```

Figure 5. A machine-coded loader program.

Monitor

The program for a monitor is given in **Fig 6**; this can be programmed using the loader of **Fig 5** by changing the address in line 10 to 14704 and inserting the new data in as many lines as necessary. Only the two character code bytes in **Fig 6** are put in the DATA lines, not the four character addresses. The monitor provides facilities which are simple but adequate for the writing and editing of hexadecimal code. A brief description of these follows.

The monitor is entered by PRINT USR 15360 and will then prompt for a 4 digit hexadecimal address to be entered. The addresses and codes will be displayed in a similar format to **Fig 6** but modified for the reduced screen size of the Spectrum. 20 rows of 8 addresses each are displayed, a flashing cursor denotes the current edit position and can be moved around by the standard cursor control keys. Code is written to memory by typing the 2 digit hex code, a warning buzz indicates invalid characters.

The use of certain non hex characters will select other facilities, these are shown in **Table 1.**

```
  1 REM Modifications  to
        loader for MONITOR
 10 LET a=14704
200 DATA "CD 74 3A 3E 08 32 6A"
210 DATA "5C CD 5C 3A DD 2A 7D"
220 DATA "5C ED 5B 7D 5C 3E 08"
900 REM Continue for the rest
        of the code
```

The routine Run and Copy will not be executed until ENTER is pressed. If a mistake has been made a system crash can be avoided by pressing any other key which will abort and return to the display. The Copy routine utilises the POKE subroutines in the Sinclair ROM in order to be compatible with the EPROM programmer previously referred to which it can directly drive. It should be noted in this context that there was a small drawing error in the circuit published in the February issue which showed pin 6 of IC3b going to A13 instead of A15, this should be corrected before running the loader programmes shown here.

Circuit Details

In the circuit of **Fig 3** the mode selection bistable is constructed by the cross connection of gates IC3a and IC3b. It is set and reset whenever pin 3 or pin 11 respectively go to logic high. Setting the bistable makes pin 8 go high which will select the EPROM mode. A set input is generated from IC1a whenever the MREQ and M1 inputs are low, address lines A14 and A15 are low and A10 to A13 are high. This situation occurs when the Z80A is reading an operation code from memory locations 3C00 to 3CFF. Once set the Sinclair ROM is disabled by the ROM CS line being pulled high via diode D1. Since TTL gates have only a limited ability to pull high the resistor R2 is added to boost the logic high output level.

The EPROM will be selected and enabled onto the data bus whenever the output of IC2a is low. This occurs when the bistable is set and a read from addresses up to 7FFF occurs, this requires that RD, MREQ, A14 and A15 are all low into IC1b. Whenever the EPROM is selected the monostable IC5 is triggered which gives a short signal on the WAIT line, this lengthens the read cycle of the Z80A allowing slower and less expensive grades of EPROM to be used. The drive to the WAIT line is via the open collector gate IC4b, this is provided to allow for other expansion boards to be simultaneously connected, provided they also have open collector drives. The bistable is reset to resume normal operation when M1, RD, MREQ, A14 & A15 are low to gates IC1b and IC3c, and at least one of A12 or A13 are low making IC4a O/P low.

```
3970  CD 74 3A 3E 08 32 6A 5C CD 5C 3A DD 2A 7D 5C ED
3980  5B 7D 5C 3E 08 32 6A 5C CD 74 3A CD 49 3B CD 74
3990  3A FD CB 01 AE CD 09 3C FD CB 01 6E 28 F7 3A 08
39A0  5C FE C8 2C 04 DD 2B 18 D6 FE C9 2C 04 DD 23 18
39B0  CE FE 0A 20 08 06 08 DD 23 10 FC 18 C2 FE 0B 20
39C0  08 06 08 DD 2B 10 FC 18 B6 FE 52 CA BA 3A FE 4D
39D0  28 A6 FE 5A CA 04 3B FE 58 28 18 CD FA 39 DD 77
39E0  00 DD 23 18 9A D5 DD E5 F5 CD 06 3C F1 CD 94 3E
39F0  DD E1 D1 C9 CD 34 3A CD 42 3A CD E5 39 CB 27 38
3A00  F3 CB 27 CB 27 CB 27 F5 CD 24 3A FD CB 01 AE CD
3A10  42 3A 18 03 CD 3F 3A CD E5 39 CB 7F 20 F6 C1 80
3A20  CD 24 3A C9 DD E5 F5 11 80 00 21 0C 01 CD 11 3C
3A30  F1 DD E1 C9 DD E5 F5 11 00 01 21 00 03 18 EE CD
3A40  34 3A 3E FF 3D DD 23 DD 2B 20 F9 FD CB 01 AE CD
3A50  09 3C FD CB 01 6E 28 EA 3A 08 5C C9 21 D5 3A CD
3A60  94 3A CD 24 3A CD F7 39 F5 CD F7 39 D1 5A 5F ED
3A70  53 7D 5C C9 3E 01 D5 DD E5 CD 15 3C 06 04 CD 1B
3A80  3C 3E 16 CD 06 3C 3E 0C CD 06 3C 3E 00 CD 06 3C
3A90  DD E1 D1 C9 E5 CD 74 3A E1 7E 23 FE 00 28 D4 E5
3AA0  CD 06 3C 18 F3 3E 06 CD 06 3C 18 ED 55 53 45 52
3AB0  20 41 44 44 52 45 53 53 20 00 21 AC 3A CD 94 3A
3AC0  CD 62 3A CD 42 3A FE 0D C2 7B 39 DD 2A 7D 5C CD
3AD0  20 3C C3 7B 39 4D 45 4D 4F 52 59 20 00 20 54 4F
3AE0  20 41 44 44 52 45 53 53 20 00 20 46 52 4F 4D 20
3AF0  41 44 44 52 20 20 00 20 45 4E 44 20 41 44 44 52
3B00  20 20 20 0C 21 DD 3A CD 94 3A CD 62 3A D5 21 EA
3B10  3A CD A5 3A CD 62 3A D5 21 F7 3A CD A5 3A CD 62
3B20  3A CD 42 3A FE 0D 20 AC DD E1 C1 ED 43 7D 5C DD
3B30  7E 00 CD 0C 3C 37 3F DD E5 E1 ED 52 CA 7B 39 03
3B40  DD E5 E1 23 E5 DD E1 18 E6 D5 3E 02 CD 16 3C 3E
3B50  16 CD 06 3C 3E 00 CD 06 3C 3E 00 CD 06 3C D1 06
3B60  14 C5 CD D3 3B 06 08 C5 DD E5 E1 7A BC 20 6A 7E
3B70  BD 20 06 1A CD E0 3B 18 04 1A CD B1 3B 13 C1 10
3B80  E6 C1 3E 20 CD 06 3C 3E 20 CD 06 3C 3E 20 CD 06
3B90  3C 10 CE C9 FE 30 FA AE 3B FE 3A F2 A1 3B D6 30
3BA0  C9 FE 41 FA AE 3B FE 47 F2 AE 3B D6 37 C9 3E FF
3BB0  C9 F5 3E 20 CD 06 3C F1 F5 CB 1F CB 1F CB 1F CB
3BC0  1F CD C5 3B F1 E6 0F FE 0A 38 02 C6 07 C6 30 CD
3BD0  06 3C C9 D5 7A CD B1 3B D1 D5 7B CD B8 3B D1 C9
3BE0  F5 3E 20 CD 06 3C F1 FD CB 55 FE CD B8 3B FD CB
3BF0  55 BE C9 00 00 00 00 00 00 00 00 00 00 00 00 00
3C00  F3 CD 70 39 FB C9 D7 F3 C9 FF F3 C9 CD 83 1E F3
3C10  C9 CD B5 03 F3 C9 CD 01 16 F3 C9 CD 44 0E F3 C9
3C20  CD 25 3C F3 C9 DD E9 0C 00 00 00 0C 00 00 00 0C
3C30  01 63 00 C9 21 00 30 11 00 80 01 00 10 ED B0 C9
```

Figure 6. Section of the monitor program showing how it can be loaded with a modified version of the loader program.

TABLE 1

M (Memory)	This is virtually a restart and a new address will be prompted and displayed.
R (Run)	Run a machine code program from the address which will be prompted.
Z (Copy)	Copy a block of code from one address to another. The beginning and end address of the data to transfer and the starting address of its destination will be prompted.
X (Exit)	Return to basic.

Figure 7. A monitor program that can be used with the board.

E&CM

SOFTLY SOFTLY

Ann Houghton has selected a number of the more interesting of readers' software submissions. We're always pleased to receive software and each program that we publish has been extensively tested, and if necessary, debugged by us. If any program fails to run, please check first for any entry errors before seeking further help.

All software that we publish will be paid for and any submissions should be sent to the following address:-

Softly Softly,
Electronics & Computing Monthly,
Scriptor Court,
155 Farringdon Road,
London,
EC1R 3AD.

Biorhythms for ZX81 16K (or ZX80 8K) by W. Hall

The program asks for the user's date of birth, and the present date. It then plots the three biorhythmic curves, Physical, Emotional and Intellectual and shows where on these cycles the user is.

The second part of the program analyses these positions and advises the user how to conduct his or her day.

```
   1 REM BIORHYTHM BY W. HALL
   5 SLOW
  10 PRINT "THIS IS YOUR BIORHYT
HM CHARTING AND ANALYSIS PROGRAM
"
  15 PRINT
  20 PRINT "PLEASE ENTER YOUR NA
ME"
  25 PRINT
  30 INPUT U$
  35 PRINT "HELLO ";U$
  40 PRINT
  45 PRINT "ENTER DAY OF BIRTH",
  50 PRINT "DAY"
  55 INPUT D
  60 IF D<1 OR D>31 THEN GOTO 55
  65 PRINT D,,"MONTH"
  70 INPUT M
  75 IF M<1 OR M>12 THEN GOTO 70
  80 PRINT M,,"YEAR"
  85 INPUT Y
  90 IF Y<1899 OR Y>1982 THEN GO
TO 85
  95 CLS
 100 PRINT "ENTER TODAYS DATE"
 105 PRINT "DAY"
 110 INPUT E
 115 IF E<1 OR E>31 THEN GOTO 11
0
 120 PRINT E,,"MONTH"
 125 INPUT N
 130 IF N<1 OR N>12 THEN GOTO 12
5
 135 PRINT N,,"YEAR"
 140 INPUT Z
 145 IF Z<Y THEN GOTO 140
 150 CLS
 155 LET A=(Z-Y-1)*365.25
 160 IF M=2 THEN LET B=334
 165 IF M>=3 AND M<=7 THEN LET B
=365-INT ((M*30.42)-30.42)+1
 170 IF M=1 OR M>=8 AND M<=12 TH
EN LET B=365-INT ((M*30.42)-30.4
2)
 175 LET C=A+B-D
 180 IF N=1 OR N>=8 AND N<=12 TH
EN LET F=INT ((N*30.4)-30.4)
 185 IF N=2 THEN LET F=31
 190 IF N>=3 AND N<=7 THEN LET F
=INT ((N*30.4)-30.4)-1
 195 LET G=F+E
 200 LET H=C+G+1
 205 LET J=H/23
 210 LET K=H/28
 215 LET L=H/33
 220 LET O=J-INT J
 225 LET P=K-INT K
 230 LET Q=L-INT L
 235 LET R=INT (O*23)
 240 LET S=INT (P*28)
 245 LET T=INT (Q*33)
 250 PRINT AT 0,6;U$,"YOU ARE IN
"
 255 PRINT AT 2,1;" ";"PHYSICAL"
 260 FOR I=0 TO 16
 265 PRINT "-";
 270 NEXT I
 275 FOR V=0 TO 22
 280 PLOT V,36+7*SIN (V/11*PI)
 285 NEXT V
 290 PRINT AT 4,1;" ","DAY ";R;"
OF 23"
 295 PRINT AT 3,R/2-1;"■"
 300 PAUSE 300
 305 POKE 16437,255
 310 PRINT AT 8,1;" ","EMOTIONAL
"
 315 FOR I=0 TO 16
 320 PRINT "-";
 325 NEXT I
 330 FOR V=0 TO 27
 335 PLOT V,25+9*SIN (V/14*PI)
 340 NEXT V
 345 PRINT AT 9,S/2-1;"■"
 350 PRINT AT 10,1;" ";"DAY ";S;
" OF 28"
 355 PAUSE 300
 360 POKE 16437,255
 365 PRINT AT 15,1;" ","  INTELL
ECTUAL"
 370 FOR I=0 TO 16
 375 PRINT "-";
 380 NEXT I
 385 FOR V=0 TO 32
 390 PLOT V,11+11*SIN (V/16*PI)
 395 NEXT V
 400 PRINT AT 16,T/2-1;"■"
 405 PRINT AT 17,1;" ","  DAY ";
T;" OF 33"
 410 PRINT AT 20,0;O;E;" ";N;" "
;Z
 415 PAUSE 300
 420 POKE 16437,255
 425 PRINT AT 20,16;"[illegible]"
 430 PRINT AT 21,16;"[illegible]
"
 435 INPUT Z$
 440 CLS
 445 LET X$="YOUR PHYSICAL CHART
 SHOWS YOU  "
 450 LET Y$="YOUR EMOTIONAL CHAR
T SHOWS YOU  "
 455 LET Z$="YOUR INTELLECTUAL C
HART SHOWS  YOU"
 460 LET A$="AT THE START -A [illegible]
[illegible] -  TAKE CARE"
 465 LET B$="IN GOOD PART OF CYC
LE - [illegible] "
 470 LET C$="AT THE TOP. YOU ARE
 NOW AT YOUR BEST"
 475 LET D$="JUST PAST THE PEAK
OF CYCLE"
 480 LET E$="AT MID-POINT-      ■
[illegible] - TAKE CARE"
```

```
485 LET F$="GETTING TO LOW PART
"
490 LET G$="AT THE [illegible] POINT
-               TAKE IT EASY"
495 LET H$=" TOWARDS THE END OF
THE CYCLE"
500 PRINT
505 PRINT
510 PRINT U$;" ON ";E;" ";N;" "
;Z
515 PRINT
520 PRINT
525 IF R=0 THEN PRINT X$;A$
530 IF R>=1 AND R<=5 THEN PRINT
X$;B$
535 IF R=6 THEN PRINT X$;C$
540 IF R>=7 AND R<=10 THEN PRIN
T X$;D$
545 IF R=11 THEN PRINT X$;E$
550 IF R>=12 AND R<=16 THEN PRI
NT X$;F$
555 IF R=17 THEN PRINT X$;G$
560 IF R>17 THEN PRINT X$;H$
565 PAUSE 400
570 POKE 16437,255
575 PRINT
580 PRINT
585 IF S=0 THEN PRINT Y$;A$
590 IF S>=1 AND S<=6 THEN PRINT
Y$;B$
595 IF S=7 THEN PRINT Y$;C$
600 IF S>=8 AND S<=13 THEN PRIN
T Y$;D$
605 IF S=14 THEN PRINT Y$;E$
610 IF S>=15 AND S<=20 THEN PRI
NT Y$;F$
615 IF S=21 THEN PRINT Y$;G$
620 IF S>21 THEN PRINT Y$;H$
625 PAUSE 400
630 POKE 16437,255
635 PRINT
640 PRINT
645 IF T=0 THEN PRINT Z$;A$
650 IF T>=1 AND T<=7 THEN PRINT
Z$;B$
655 IF T=8 THEN PRINT Z$;C$
660 IF T>=9 AND T<=15 THEN PRIN
T Z$;D$
665 IF T=16 THEN PRINT Z$;E$
670 IF T>=17 AND T<=23 THEN PRI
NT Z$;F$
675 IF T=24 THEN PRINT Z$;G$
680 IF T>24 THEN PRINT Z$;H$
685 PAUSE 400
690 POKE 16437,255
695 PRINT AT 19,0;"PRESS CONT A
ND [illegible] TO CONTINUE"
700 PRINT
705 PRINT "  OR [illegible] TO END"
710 INPUT O$
715 CLS
720 IF CODE O$=40 THEN GOTO 5
725 IF CODE O$<>40 THEN PRINT A
T 0,6;"[illegible]"
730 PAUSE 100
735 POKE 16437,255
740 FOR J=3 TO 20
745 PRINT AT J,9;"[illegible]"
750 NEXT J
755 STOP
```

"LOAN" by John Allsebrook (1K)

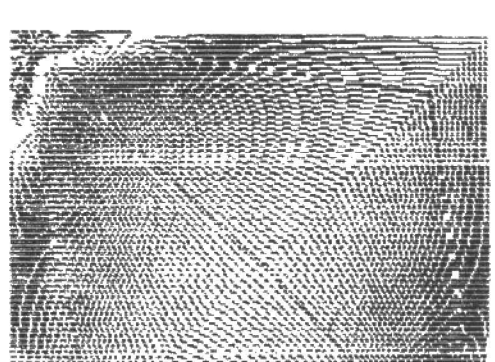
This program calculates mortgage repayments, and is very useful in view of the changing interest rates.

```
2 PRINT AT 11,1;"HOW MUCH DO
YOU WISH TO BORROW"
3 INPUT Y
4 CLS
5 PRINT AT 11,1;"HOW MANY YEA
RS TO REPAY"
6 INPUT N
7 CLS
8 PRINT AT 11,1;"WHAT IS PRES
ENT MORTAGE RATE"
9 INPUT A
10 CLS
11 LET B=A/100
12 LET C=B*Y
13 LET D=(1+B)**N
14 LET E=1-D
15 LET F=E/D
16 LET P=C/F
17 PRINT AT 4,6;"£";P;" ";"PER
ANNUM"
18 LET M=P/12
19 PRINT AT 6,6;"£";M;" ";"EAC
H MONTH"
20 GOTO 1
```

Spectrum Stocking Filler

Spend a few minutes keying in this short program and watch amazing patterns appear in glorious colour before your very eyes!

```
10 CLS : BRIGHT 1: OVER 1: BOR
DER 0: PAPER 0: INK 4
20 LET X=175
30 FOR I=-X TO 0 STEP RND+1
40 GO SUB 100
50 NEXT I
60 GO SUB 400
70 CLS : INK RND*6+1
80 GO TO 30
100 PLOT 200,0
110 DRAW I,X+I
120 PLOT 24,X
130 DRAW X+I,I
140 RETURN
400 FOR I=1 TO 7
410 FOR J=0 TO 21
420 PRINT PAPER I;AT J,3;"
                         "
421 REM          23 SPACES
430 NEXT J
440 PAUSE 25
450 NEXT I
460 POKE 23692,50
470 PRINT ''''''''''''''''''''''
471 REM     21XSYMBOL SHIFT 7
480 RETURN
```



Hi-Res Computer Project

Our high performance, 6809 based, computer project started publication back in October 1982. Gary Evans, E&CM's 'New Boy', outlines the story to date for the benefit of our many new readers, and for those of you *au fait* with the computer, has news of how the design will develop over the next few months.

To publish a series of articles describing the construction of any computer system is a fairly ambitious undertaking. To publish the design of a sophisticated hi-res graphics computer, supporting the FLEX-9 operating system, goes beyond the constraints set by the majority of computing titles. Electronics & Computing Monthly however, started publishing just such a design over eight months ago.

The aims of the project as outlined by John Chewter, the systems designer, in October 1982 are set out in **Table 1** and these six design goals have been maintained throughout the development of the project.

The first article described the heart of the system, the 6809 CPU card. The design for the card is based on the old, in computing terms, Newbear 77-68 system. The rights to the Newbear system had been acquired by Stirling Microsystems who agreed to allow us to utilise the 77-68 bus structure for our system (this has meant that the large number of people with 77-68 systems have been able to use our new hardware with their old system and vice versa).

Another important aspect of the system introduced in the first article, was that it would be capable of supporting the FLEX-9 operating system – the industry standard single user 6809 operating system.

Reprints of all articles to date are available from our offices at a price of £2.95 fully inclusive.

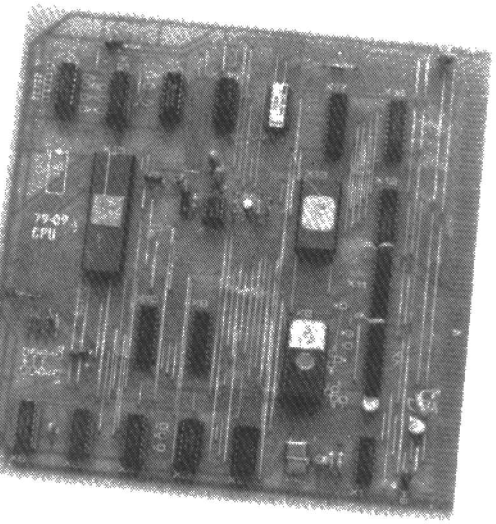


TABLE 1

- ★ To design the system around Hitachi's HD 6809 MPU, one of the most advanced 8 bit processors.
- ★ To design a truly flexible system that can grow with the user.
- ★ To provide a multi-language machine capability.
- ★ To utilise an established operating system supporting existing, high performance software.
- ★ To incorporate the facility for further expansion and change of MPU if the need arises.
- ★ To keep the cost as low as possible without degrading performance.

Hardware – Month One

The CPU card is capable of operation as a free standing microcomputer, with the addition of a teletype etc. (suitable for industrial control or teaching applications) and consists of the HD6809 MPU, 1K of RAM, 2K of EPROM based monitor, together with parallel and serial input/output ports, baud-rate generator and bus buffer drivers. The board is compatible with SWTP's 6809 software and use of their SBUG-E 1.5 will enable the card to be used with a teletype in the stand alone applications mentioned above.

The CPU card is, in common with the rest of the system, designed for mounting in a standard Vero Rack cage. The physical layout of the cards varies slightly from the 77-68 bus in that all I/O lines are brought out from the front of the cards (the exception being the CPU card itself which may be modified to comply with this scheme). The reason for this departure from the 77-68 format is to enable all boards to be interchangeable and to free some backplane lines so that expansion to a 68000 (or similar) based system is possible in the future.

Month Two

The second part of the series (November '82) began with detailed information concerning the construction of the CPU card before moving on to describe an exhaustive test procedure of the board using S-BUG 1.5 and a dumb terminal/teletype.

This part of the series then went on to describe a versatile CUTS cassette interface based on the Cottis-Blandford high speed design. The card is capable of operation at 2400, 1200, 600 and 300 baud. The provision of the 300 baud option is particularly important as this is

the speed at which most commercial CUTS software is produced.

The article outlined the principle of the CUTS recording format and gave a full description of the circuit as well as testing procedures and instructions on using the card to dump and load programs.

Third Time Around

Part three of the series (December '82) began by clearing up some confusion as to what FLEX-9 is all about. John Chewter explained that FLEX-9 is an operating system that controls floppy disc drive(s), formats new discs, copies one disc to another, keeps a disc's catalogue, supports editing facilities for producing assembler source programs, contains a 6809 assembler plus many other utilities.

From the above it should be evident that FLEX-9 is **not** a language *a la* BASIC, but forms the heart of a floppy based system which can load BASIC or for that matter any other language. The power of the FLEX-9 OS cannot be realised until floppies are added to a system – there being no OS as such until this time, the computer being run by the system bug (monitor).

A small program to exercise the CPU card was then described before the article moved on to the major topic of the month – a 64K dynamic RAM card.

The RAM card is based on HM4864-3 64K devices and is refreshed transparently. A full circuit description of the card, together with the construction and testing procedures were given in the December article.

This part of the series concluded by pointing out that with the cards so far described and a cassette based BASIC interpreter, a powerful low-cost computer system was at the constructor's disposal.

Into '83

January of this year saw part four of the project published, this concerning itself with a description of the system's VDU card. The sophisticated nature of the VDU card is one of the major attractions of the E&CM computer.

The VDU has a 512 x 512 pixel array in eight colours with two pages of information stored on the video board. This is an important element of the design – the VDU memory is separate from the main system's RAM and thus there are none of the screen/program memory conflicts encountered on many micros. The heart of the card is the EF 9365 Graphic Display Processor (GDP) from Thompson CSF. This is a very powerful device whose main attributes are set out in **Table 2.**

TABLE 2

★ 512 x 512 or 256 x 256 graphics.
★ High speed vector plot (up to 1,500,000 dots/second).
★ Full ASCII character set on chip.
★ Performs all refreshing for display RAM and for transparently updating the display.
★ Line drawing by hardware.
★ Light pen capability.
★ Programmable ASCII character size.
★ ASCII characters may be written horizontally or vertically.
★ Character may be sloped to provide alternate fonts.
★ Lines may be continuous, dotted, dashed or dashed and dotted.

Part Five

The next article in the series (February '83) carried on with an overview of the hardware for the VDU card while part six in April (we gave John Chewter a month off in March) gave full circuit diagrams and descriptions of the various sections of the VDU card.

As described, the VDU card provides both RGB and sync outputs and a composite signal for monochrome operation. It is planned to describe the connection of a UHF modulator at a later stage.

To date and Beyond

Well even our newest readers will probably have seen the May issue but just in case you haven't, the article described some software designed to put the system to date through its paces.

At the start of this article we mentioned the early involvement of Stirling Microsystems. Stirling have been supplying kits of parts for each of the cards published in the series to date. From next month their role will change slightly for although they'll still be supporting the series with hardware kits they'll also be taking a more active part in the design of the computer. Hardware for the system already includes the eagerly awaited disc controller card (to be published next month) with designs for A/D and D/A, I/O and PROM programmer cards well under way.

The plan is to publish details of hardware and software for the system in alternate months. Thus the Disc Controller in July will be followed by a description of the FLEX OS which will come into its own when this card is added to the system. The A/D, D/A board will in similar fashion be followed by an article describing applications for this board.

We've pencilled in plans for as far ahead as February 1984 which has 68000 W-processor against it . . . it'll be worth the wait.

Some Useful Addresses

Stirling Microsystems,
241 Baker Street,
London,
NW1 6XE.
Suppliers of complete kits of parts for the various system cards.

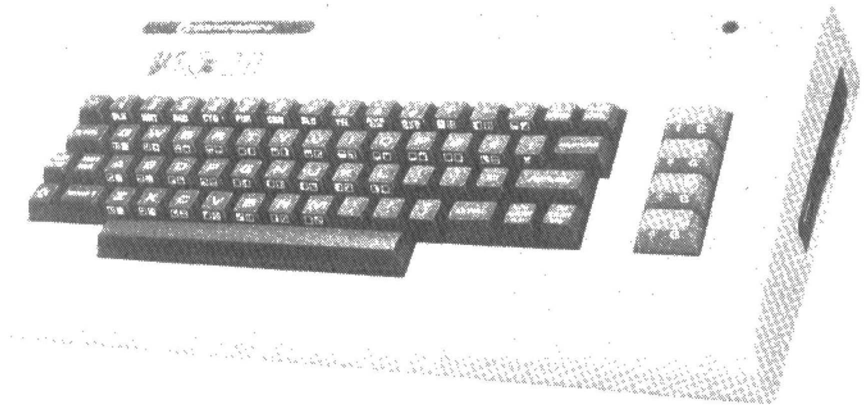
68 Group,
41 Pebworth Road,
Harrow,
Middlesex,
HA1 3UD.
Membership £5 – the main user group for the system and one that incorporates the old 77-68 group. The group is for all 68xx micro users.

Teddy Technical,
Electronics & Computing Monthly,
Scriptor Court,
155 Farringdon Road,
London,
EC1R 3AD.
For the technical queries on magazine articles. Replies will only be via the columns of E&CM.

E&CM

Vic 20 Software

We've looked at a selection of the wealth of software for both the standard and expanded memory Vic 20.



The Vic 20 is fast becoming outdated with the advent of more sophisticated computers, and as a consequence is rather slow in loading programs from cassette.

On the whole the software reviewed has made good use of the colour, graphics and sound of the Vic 20, but at times more attention could have been paid to detail, for example on screen instructions are sometimes difficult to read because of the colours used, and the words being rather cramped together. There were also one or two spelling mistakes.

As the Vic 20 has its own cassette recorder, there are no problems in setting volume levels when loading programs, and on the whole program loading was trouble free. Cartridges, of course, load instantly.

VIC INVADERS for unexpanded Vic 20 from BRIDGE SOFTWARE, 36 Fernwood, Marple Bridge, Stockport, Cheshire SK6 5BE, **Price £6.90.** This variation of the popular arcade game gives 5 skill levels, which set the number of bases, and also the speed at which the invaders advance. A high score is recorded at any one skill level. I did not find this game required as much skill as other invaders games I've seen, even at the most difficult level. The base moves very quickly from side to side using the cursor keys, and the space bar fires the laser.

The display was clear, and the sound effects good, but the only way to see when a base has been hit is to look at the number of bases you have left at the top of the screen.

THE VIC MULTISOUND SYNTHESIZER for unexpanded Vic 20 from ROMIK LTD., 24 Church Street, Slough, SL1 1PT, **Price £9.99.** Create fantastic sound effects from your keyboard. Music can be played like on a normal organ or from a synthesizer. Tunes can be programmed and played back with a background of drums, or saved to tape. The computer keeps the background and drum beat in time with one another, and you can play another tune over the top at the same time. There are numerous combinations of sounds available – changes of octave, speed, temp of drums, vibrato, decay etc.

At first I found programming tunes to replay tricky, and got in a muddle in putting them into one of the four memories, but having mastered that, and sorted out what notes were obtained from the keyboard, I managed to make some interesting variations on well known tunes. This program has great potential for making original music from the computer.

PATIENCE for Vic 20 +3K or over from C. P. WHITE (SERVICES), 52 Northfield Avenue, West Ealing, London W13 9SY, **price £3.50.** If you find shuffling cards and laying them out tedious, then this could be the game for you. It provides a realistic game of Patience, giving you a score in keeping with the number of cards you can get "out". The program gives a full card display, and gives options to place a card into play, suit up boxes, select card from the pack etc. No cheating is allowed, as illegal inputs are ignored!

The one irritation I found in an otherwise well presented game was that when you are asked for your move, the card you select is displayed in either red or black, but not necessarily its correct colour, for example the Ace of Spades could be shown in red. I found this confusing at first, but got used to it after a while.

The sound effects allow you to hear the shuffling and laying of the cards, and added to the enjoyment and realism of the game.

ANOTHER VIC IN THE WALL from BUG-BYTE SOFTWARE, 98-100 The Albany, Old Hall Street, Liverpool L3 9EP, **for unexpanded Vic 20, price £7.00.** By controlling a bat with either joystick or keyboard, you attempt to catch a ball which bounces off bricks at the top of the screen. The rows of bricks increase a row at a time, coming lower down the screen and giving you less time to catch the ball.

At first the game seems very slow, but as it progresses it gets faster requiring quick reflexes to avoid dropping the ball.

The game looked relatively easy when watching its demonstration, but the computer's skill is obviously far greater than mine!

I enjoyed this game very much, but found it easier to control with the keyboard than with the joystick.

THE WIZARD AND THE PRINCESS for unexpanded Vic 20 from MELBOURNE HOUSE, Glebe Cottage, Glebe House, Station Road, Cheddington, Leighton Buzzard, Beds LU7 7NA, **price £5.95.** This is an adventure game in 5 parts, in which you have to find and enter the evil wizard's castle, slay the dragon guarding the gate, find your way through the maze and find the princess and rescue her. Finally you must escape from the castle with the princess.

I was able to find the castle easily and slay the dragon, but the maze defeated me and I had to use the option to pass on to the next chapter.

I then encountered the wizard who I successfully killed, and rescued the princess. Escaping from the castle and killing the troll who guarded the bridge was relatively easy.

At each stage of the game I had to refer to the instructions to see which keys to use, as these are different for each stage.

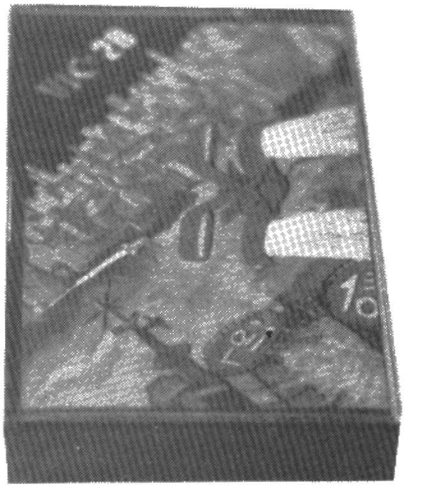
When the rescue has been completed, the game has to be reloaded before it can be played again.

STAR WARP II for Vic with 16K expansion memory from SOFT TOYS, 14 Lockharton Avenue, Edinburgh EH14 1AZ, **Price £7.** You are the captain of a ship, your mission being to explore space. You can command one of a selection of ships with varying numbers of crew members, firing strength, fuel levels etc.

During your exploration, you encounter many different types of alien, and decisions have to be made on how to deal with them – whether to use torpedoes or phasers, how much power to allocate to your shields, whether to retreat or attempt to board the alien vessel. Details of the strength of the vessel are shown on screen.

On reaching an unknown planet, information is given on its resources and inhabitants, and you must decide what action to take in order to restock your ship.

There are many aspects to this game making it interesting and absorbing. It requires quick decisions and actions to avoid having serious damage inflicted on your ship by the aliens, and having your resources drained by taking the wrong action.



STARSHIP ESCAPE for Vic 20 with 16K expansion from SUMLOCK ELECTRONIC SERVICE (M/CR) LTD., Royal London House, 198 Deansgate, Manchester M3 3NE, **price £9.95.** In this game the stellar scout ship of which you are the captain is drawn by a tractor beam into an alien ship. Your ship is then disassembled and the parts scattered throughout various rooms of the alien ship. Your task is to locate the various parts and reassemble your ship, by moving from room to room of the alien ship.

During your search, you encounter various hazards, androids, spiders, alien beings and the sinister Force Cloud Entity.

This game can be controlled by keyboard or joystick (I actually found it easier with the keyboard) and is extremely tricky to play because of the high speeds at which the enemies move. I did, however, manage to collect some of the parts, but tended to get stuck in the doorways with them, and had to drop them and pick them up differently.

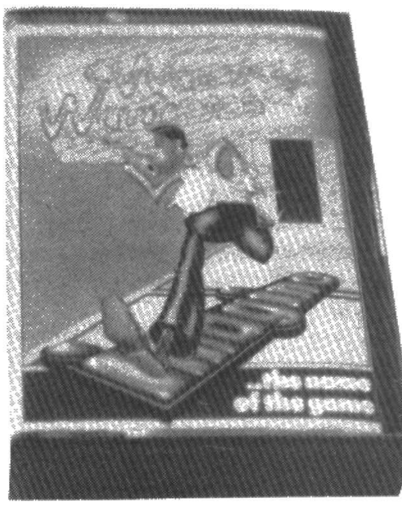
I then encountered the Force Cloud Entity which invariably caught me, and took the parts from me, and I was returned to the airlock to start again.

CHOPLIFTER – Cartridge for standard Vic 20 for use with joystick from CREATIVE SOFTWARE, distributed by AUDIOGENIC, PO Box 88, Reading, Berks. **Price £24.95.** An exciting, well presented game in which the object is to rescue American hostages by helicopter, and return them to the U.S. base. You must pilot your helicopter past tanks, jet aircraft and satellites to get to the hostages. The direction of the helicopter is controlled by the joystick, turning by means of the fire button, and firing by pressing and releasing the fire button very quickly.

Points are scored for every hostage safely returned to the base, which can prove difficult if you are unlucky enough to come under fire from the enemy.

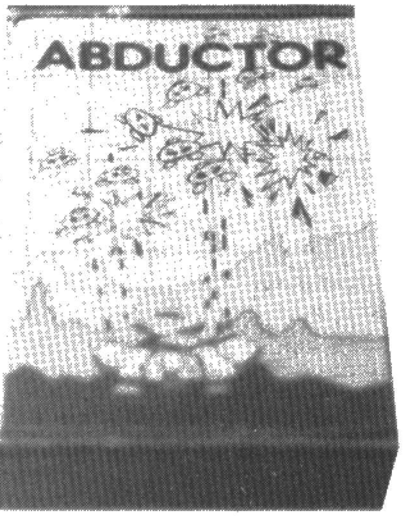
This is a very good game requiring much skill and concentration. The graphics are excellent and the sound effects very realistic.

WACKY WAITERS for unexpanded Vic 20 from IMAGINE SOFTWARE, Masons Building, Exchange Street East, Liverpool L2 3PN, **price £5.50.** If you're thinking of getting a job as a waiter, think carefully – it's not as easy as you might think!



The object of this game is to wait for a guest to call for service, find out what they require and fetch it for them. In order to reach the guest you must use a series of lifts, and then return via the lifts to collect the order. Your trip is dependant on how quick you are. If you trip getting into the lift you get a warning letter from the manager who is watching for such occurrences. Too many warning letters result in the sack! If your anxiety to please both guest and manager cause you to hurry too much when getting into one of the lifts, there is a long drop to the bottom of the lift shaft, and a sudden end to your career.

This game required skill and concentration, and could be played with either joysticks or keyboard, and was well presented and absorbing.



ABDUCTOR for unexpanded Vic from LLAMASOFT SOFTWARE, 49 Mount Pleasant, Tadley, Hants RG26 6BN, **Price £6.** A joystick is essential for this game, as provision is not made for keyboard control. It is a very fast moving game involving the shooting of Alien Nasties who are attempting to carry off your men. If they succeed in carrying off a man, they throw his skull back, and if you don't manage to dodge it, you lose a life. If you hit the alien while he is carrying off the man, the man will float safely back down to the surface.

Unfortunately, the men at either side of the screen cannot be rescued as the cannon will not reach that far over.

DESIGN AND RESTORE for unexpanded Vic 20 from KAYDE ELECTRONIC SYSTEMS LTD., The Conge, Great Yarmouth, Norfolk NR30 1PJ, **price £9.99.** This program enables you to create your own graphics which can be loaded into RAM for use with your own programs. Options are given to create characters, load a previously saved character set, display characters in different positions, and save to tape your newly created character set. Restore is a small program which will load a character set into RAM.

Unfortunately I found it difficult (perhaps due to a lack of instructions with the review cassette) to edit quickly. I found only three keys for the setting up of graphic characters, and there did not appear to be any way to backspace if an error had been made.

GRIDDER for unexpanded Vic 20 from TERMINAL SOFTWARE, 28 Church Lane, Prestwich, Manchester, **price £7.95.** This is a maze chase game in which you are in control of a painter on a grid, attempting to complete squares of the grid. If you are successful, the square changes colour and you score 10 points. The grid must be completed within a set time, and extra points are awarded for quick times.

Also occupying the grids are chasers who attempt to catch you. Every time you are caught you lose a life (you start with 3).

Each grid is different, and they get progressively more difficult. You can make a hole in the grid which neither you nor the chasers can cross, by pressing the fire button on the joystick. The game can be played from the keyboard, but the keys used are very close together making control difficult.

The display is easy to see and well presented, making this a very engrossing game.

MASTERMIND for 8K or 16K Vic 20 from VICSOFT, 818 Leigh Road, Trading Estate, Slough, Berks, **price £9.99.** This is a version of the BBC quiz contest in which 1-4 players can partake. The first round is a specialised subject chosen by the contestants and each contestant has two minutes on each subject. The data cassette included with this program has science/technology questions and answers on one side, and general knowledge on the other. Other specialist subject and general knowledge cassettes are available at £1.99 each.

Clear instructions are given on how to load and play the game. If contestants choose different subjects from one another, the data has to be loaded from cassette during the game. Once the questions have been loaded, a countdown to the start of the two minutes commences. As with the television programme it is better to "pass" a question rather than waste time trying to remember the answer.

The program is introduced by a welcome and the mastermind theme, and is interesting and enjoyable. The only problems, which are unavoidable, are that the data cassettes take between 1½ and 2 minutes to load, and the first contestant is at a disadvantage because later contestants may have seen some of the questions before and may remember some of the answers.

A COUNTRY GARDEN for Vic 20 with at least 8K memory expansion from AUDIOGENIC LTD., P.O. Box 88, Reading, Berks, **price £7.95.** Your garden is being infested by caterpillars, snails and fleas, and mushrooms are springing up all over the lawn. You are in control of an aggressive mole which is controlled either by the joystick or the keyboard.

You have four moles which fire at the pests scoring points for those it hits. After every four caterpillars, the mushrooms are regenerated, and you score 10 points for every mushroom left on the screen.

Your current score and high score are displayed, but unfortunately the game failed to restart after the 4 moles had been destroyed, so the game had to be reloaded and the high score was not displayed, which spoilt an otherwise amusing and novel game.

Acknowledgements

Our thanks in preparing the Software Review go to the following:

For supplying the Vic 20 and cassette recorder, Peter Walker Associates (on behalf of Commodore Business Machines); The 16K RAM pack used by courtesy of Plus 80 Ltd.; Joystick kindly provided by Kempston Microelectronics.

If you're still staggering through the computer jungle and not getting sensible answers to your questions, we have some good news:

Now you can turn to **professional** people who are capable of giving you sound advice on practically all aspects of popular computing.

They all have one thing in common:

They are **COMPUTERS FOR ALL** dealers.

A Computers for All dealer is different from the normal Computer retailer.

Not surprisingly he won't try to sell you things like cameras or cosmetics; stationery or sealing wax.

He will, however, be capable of answering sensibly almost any question you have on computers and computing, and have readily available a wide range of popular computers, hardware, software, books, and peripherals.

So why not call in at your local **COMPUTERS FOR ALL** dealer today? He can lead you in the direction you want to travel.

The shops where people matter:

AVON
JADE COMPUTERS
Coombend, Radstock, Bath. 0761 32570.
***MERCATOR COMPUTER SYSTEMS**
3 Whiteladies Road, Clifton, Bristol. 0272 731079.
MOBILE MICROS
2 Castle Street, Thornbury, Bristol. 0454 418383.

BERKSHIRE
***KENNETH WARD COMPUTERS**
Verve House, London Road, Sunningdale. 0990 25025.

CHANNEL ISLANDS
MEGA LTD.
7 Anley Street, St. Helier, Jersey. 0534 72263.

CORNWALL
***FAL-SOFT COMPUTERS**
8 St. George's Arcade, Falmouth. 0326 314663.

DEVON
***BITS & BYTES**
44 Fore Street, Ilfracombe, N. Devon EX34 9JD. 0271 62801.
COMPUTER SYSTEMS (TORBAY)
Pump Street, Brixham. 08045 6565/6.
***CRYSTAL COMPUTERS**
209 Union Street, Torquay. 0803 22699.

DORSET
DENSHAM COMPUTERS
329 Ashley Road, Parkstone, Poole. 0202 737493.
***WEYTECH COMPUTER SYSTEMS**
20 St. Edmunds Street, Weymouth. 03057 79881.

ESSEX
***AKHTER INSTRUMENTS**
Unit 19, Arlinghyde Estates, South Road, Harlow. 0279 412639.
***COMPUTERS FOR ALL**
72 North Street, Romford. 0708 752862.
***EMPRISE LTD.**
58 East Street, Colchester 0206 870353.
***EMPRISE LTD.**
3a Baddow Road, Chelmsford. 0245 356834.

HANTS
HOME COMPUTER CENTRE (ROSS RECORDS)
13 Kingston Road, Portsmouth. 0703 819515.
MICRO VIDEO STUDIOS LTD.
17 Turks Street, Alton. 0420 82055.

HERTS
VIDEO CITY COMPUTERS
45-47 Fishers Green Road, Stevenage. 0438 53808.

KENT
***APHROS SOFTWARE CO.**
83 Canterbury Road, Westbrook, Margate. 0843 23627.
***MEDWAY COMPUTERS**
141 New Road, Chatham. 0634 826080.
***MICRO MAGIC**
128 Erith Road, Bexleyheath. 0322 523052.

LEICS
DIMENSION
27-29 High Street, Leicester. 0533 57479.

LONDON
EUROCALC
224 Tottenham Court Rd., London W1. 01-631 4139.
KAYDE HOME COMPUTERS
1 Station Approach, New Eltham. London SE9. 01-859 7505.
***KELLY'S COMPUTERMARKET**
227 Dartmouth Road, Sydenham, London SE26 4QY. 01-699 4399/6202.

MIDDLESEX
MAYFAIR COMPUTER SERVICES
6 Main Avenue, Moor Park, Northwood. 09274 20664.
***TWILLSTAR COMPUTERS**
17 Regina Road, Southall. 01-574 5271.

N. IRELAND
D. V. MARTIN LTD.
13 Bridge Street, Belfast, BT1 1LT. 0232 226434.

NORFOLK
***CARLTON COMPUTERS**
4 Swanstons Road, Great Yarmouth, NR30 3NQ. 0493 58898.

SHETLANDS
***TOMORROW'S WORLD**
Esplanade, Lerwick, ZE1 0LL. 0595 2145.

SUFFOLK
BECCLES COMPUTERS
5 The Ridings, Worlingham, Beccles. 0502 715061.

SURREY
***ANIROG COMPUTERS**
8 High Street, Horley. 02934 2007.
COMPUTASOLVE
8 Central Parade, St. Marks Hill, Surbiton. 01-390 5135.

SUSSEX
THE COMPUTER CENTRE (BMS) LTD.
37d & 37e Robertston Street, Hastings, East Sussex. 0424 439190.

S. WALES
AUTOMATION SERVICES (S. WALES)
3 Wermeys Road, Penysfai, Bridgend. 0656 720959.
DAN EVANS (BARRY) LTD.
81 Holton Road, Barry, South Glamorgan. 0446 734242.
MICRO-CARE COMPUTING
18 Hawarden Road, Newport, Gwent. 0633 278040.
***MORRISTON COMPUTER CENTRE**
37 Clase Road, Morriston, Swansea SA6 8DS. 0792 797572.
STEVE'S COMPUTER CO. LTD.
Castle Arcade, Cardiff. 0222 41905.

WARWICKSHIRE
***IMPULSE MICRO SYSTEMS LTD.**
6 Central Chambers, Cooks Alley, Wood Street, Stratford-upon-Avon. 0789 295819.

WEST MIDLANDS
JBC
200 Earlsdon Ave. North Earlsdon, Coventry. 0203 73813.

WORCESTERSHIRE
***DEATH VALLEY COMPUTERS**
P.O. Box 54, Worcester WR2 6QA. 0905 640400.
EVESHAM COMPUTER CENTRE
Crown Court Yard, Bridge St., Evesham. 0386 48635.

COMPUTERS FOR ALL